(毎月1回1

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

Oh!X

10
1993

SOFT BANK オー！エックス
特別定価800円

SHARP
目の付けどころが、
シャープでしょ。
夢の、頂きへ。
68ワールドの最高峰。
X68030
32bit PERSONAL WORKSTATION

32ビットパーソナルワークステーション

演算速度4.3倍(当社10MHz機比)/2.4倍(当社XVI比)※1、動画ウィンドウに見る新創造次元。選ばれた人だけが持つ感性によってX68030の扉はひらかれる。

X68000シリーズとして初の32ビットMPU MC68EC030を搭載して高速化を実現。

データキャッシュ、プログラムキャッシュをそれぞれ256バイト搭載したクロック周波数25MHzの高速32ビットMPUを搭載。演算速度は2倍以上(当社従来比)※1の高速化を実現しました。また数値演算プロセッサMC68882※2(25MHz)もサポート。大量の実数演算を必要とするクリエイティブワークやGUI環境の操作性など、実行速度の飛躍的な向上が図られています。(当社従来比)

※1 Dhrystn(四則演算)比。25MHz・データキャッシュオン・プログラムキャッシュオンでMC68000/10MHz時の約4.3倍、16MHz時の約2.4倍。

※2 数値演算プロセッサCZ-5MP1標準価格54,800円(税別):本体内の専用ソケットに取りつけ可能。

65,536色表示、動画表示を実現。さらにパワーアップしたSX-WINDOWver.3.0。

X68000独自のウィンドウシステムとして定評の「SX-WINDOWver.2.0」をさらに強化した「SX-WINDOWver.3.0」を標準装備。新たに、65,536色の自然色グラフィック表示を可能とした『グラフィックウィンドウ』※を搭載。またアニメーション動画をウィンドウ上で表現でき、手軽にコンピュータアニメーションが楽しめる『CGAウィンドウ』、さらに従来のエディタのイメージを一新、高度な日本語文書作成をサポートするSX-WINDOW対応の高機能日本語マルチフォントエディタを標準装備。アウトラインフォントの展開もさらに高速化が図られています。

※SX-WINDOW上の512×512ドットのエリア内で表示可能。

GUIに対応する大容量メインメモリを搭載。

メインメモリは標準で4Mバイト、複数のアプリケーションをウィンドウ上で同時に使用するなど大量のデータ処理に対応。また本体内の増設で、I/Oスロットを使用せず最大12Mバイトまで拡張できます。拡張したメモリはすべて32ビットバスによる高速アクセスが可能、優れた拡張環境でシステムパワーアップをサポートします。

※メモリ増設には、4MB内部増設RAMボードCZ-5BE4標準価格54,800円(税別)、4MB増設RAMモジュールCZ-5ME4標準価格49,800円(税別)をご使用ください。なおCZ-5ME4はCZ-5BE4上に装着します。

X68000シリーズの高機能を継承した上で、さらに使いやすさの向上を図ったコンパチビリティ重視設計※1、すぐに使える高機能ソフトを標準装備。

●25MHzでは速すぎるアプリケーションも、従来のクロック周波数(10MHz/16MHz)で動作可能なソフトコンパチ重視設計●65,536色同時発色の自然色グラフィックス(最大表示エリア512×512ドット)、1024×1024ドットの実画面エリアを持つ高解像度表示能力(最大表示エリア768×512ドット・カラー液晶ディスプレイ使用時※2は640×480ドット)、疑似高解像度スーパーインポーズ(インターレース方式/512×512ドット・専用ディスプレイテレビ使用時)を装備した高精細度自然色グラフィックス機能。●外部MIDI音源もコントロール可能※3、ウィンドウ上で手軽にコンピュータミュージックが楽しめるMIDI音源対応デバイスドライバ搭載●ステレオ8オクターブ8重和音FM音源、ADPCM搭載●プリンタ、RS-232C、SCSI、オーディオ入出力、イメージ入力など多彩なインターフェイスを装備。●日本語変換効率や操作性を高めた日本語フロントプロセッサASKver3.0搭載。●従来のエディタのイメージを一新したSX-WINDOW対応の高速多機能日本語マルチフォントエディタ標準装備●日本語マルチフォントエディタ中に貼り付ける絵やグラフなどが簡単に作成できるグラフィックパターンエディタ●MIDI対応のX-BASIC。

※1 アプリケーションソフトおよび周辺機器のうち、一部動作しないものがあります。詳しくはシャープお客様相談窓口にお問い合わせください。

※2 10.4型カラー液晶ディスプレイLC-10C1-H標準価格598,000円(税別)、接続ケーブルAN-1515X標準価格4,200円(税別)をご使用ください(SX-WINDOW対応アプリケーションのみ。色数に制限があります)。

※3 別売のMIDIインターフェイスが必要です。

5.25"FDDマンハッタンシェイプシリーズ

■X68000伝統のマンハッタンシェイプを継承 ■5.25インチFDD2基搭載 ■80MBハードディスク内蔵(CZ-510C)※ ■マウス・トラックボール標準装備 ■ASCII準拠フルキーボード採用

※CZ-500Cには、2.5インチ80MB内蔵用ハードディスクドライブCZ-5H08/2.5インチ160MB内蔵用ハードディスクドライブCZ-5H16を用意しています。

本体+キーボード+マウス・トラックボール
5.25インチFDDタイプ CZ-500C-B(チタンブラック)標準価格398,000円(税別)
HDタイプ CZ-510C-B(チタンブラック)標準価格488,000円(税別)
14型カラーディスプレイ
CZ-608D-B(チタンブラック)標準価格94,800円(税別・チルトスタンド同梱)

3.5"FDDコンパクトシリーズ

■32ビットのハイパワーを凝縮したコンパクトフォルム ■2DD対応3.5インチFDD2基搭載 ■80MBハードディスク内蔵(CZ-310C)※ ■マウス標準装備 ■コンパクトキーボード採用

※CZ-300Cには、2.5インチ80MB内蔵用ハードディスクドライブCZ-5H08/2.5インチ160MB内蔵用ハードディスクドライブCZ-5H16を用意しています。

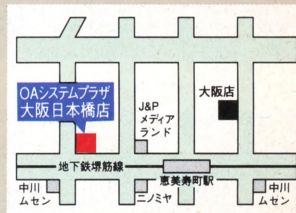
Compact

NEW

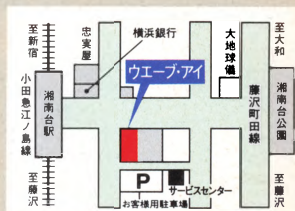
本体+キーボード+マウス
3.5インチFDDタイプ CZ-300C-B(チタンブラック)標準価格388,000円(税別)
HDタイプ CZ-310C-B(チタンブラック)標準価格478,000円(税別)
14型カラーディスプレイ
CZ-608D-B(チタンブラック)標準価格94,800円(税別・チルトスタンド同梱)

OAシステムプラザ
〈山下 章〉
アミューズメント・トーク

●日時●
9月26日(日)13:00〜16:00
●会場●
㈱OAシステムプラザ大阪日本橋店3F
大阪市浪速区日本橋5-9-16
問い合わせ先
㈱OAシステムプラザ☎(06)646-3169

(山下氏による ゲーム操作方法など)

ウェーブアイ
X68030 FORUM

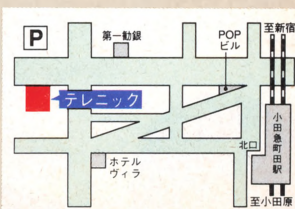
●日時●
10月2日(土)・3日(日)10:00〜20:00
●会場●
㈱ウェーブアイ湘南台店
神奈川県藤沢市湘南台1-10-1
問い合わせ先
㈱ウェーブアイ湘南台店☎(0466)43-1771

(X68超特価、下取セール実施 SX-WINDOW体験フェア)

日本インコム
X68激売フェア

●日時●
10月2日(土)・3日(日)10:00〜19:00
●会場●
㈱日本インコム テレニック町田店
東京都町田市森野1-34-13 町田薬品ビル2F
問い合わせ先
㈱日本インコム テレニック町田店☎(0427)24-2881

(SX-WINDOWの実体験 本体・周辺・ソフト超目玉品販売他)

■お問い合わせは… シャープ株式会社 コンシューマーセンター 西日本相談室 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

資料請求券 X68030 Oh!X 10係

特別企画　秋祭りPRO-68K

コットン

The World of X68000

TOWERJACK

THE USER'S WORKS

DōGA CGアニメーション講座

Oh!X

C O N T

●特別企画

●カラー紹介

●THE SOFTOUCH

〈スタッフ〉
●編集長／前田　徹　●副編集長／植木章夫　●編集／山田純二　豊浦史子　高橋恒行　●協力／有田隆也　中森　章　林　一樹　吉田幸一　華門真人　吉田賢司　朝倉祐二　大和　哲　村田敏幸　丹　明彦　三沢和彦　長沢淳博　司馬　護　石上達也　柴田　淳　瀧　康史　横内威至　進藤慶到　●カメラ／杉山和美　●イラスト／山田晴久　江口響子　高橋哲史　川原由唯　●アートディレクター／島村勝頼　●レイアウト／元木昌子　ADGREEN　●校正／グループごじら

表紙絵：須藤　牧人

E　N　T　S

1993 OCT.
10

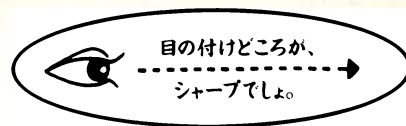
■広告目次

先が、面白くなる。

ウィンドウ環境のプラットホームを確立、SX-WINDOW ver.3.0

●この画面は広告用に作成した、機能を説明するためのイメージ画面です。また、各種アイコン等は、SX-WINDOW ver.3.0がもつ機能を使って作成したもので、標準装備のものとは異なるものもあります。
●本広告中のエディタで表示している文字のフォントはZeit社の、「書体倶楽部」のフォントを使用しています。

シャープ株式会社 ●お問い合わせは…コンシューマーセンター西日本相談室 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

資料請求券 ペリフェラル Oh!X 10条

に見たGUIの新展開。

❶マルチフォントエディタ編集例。文字ごとに文字種、文字の大きさの指定、修飾が可能で、イメージデータの貼り付けもOK。

❷CONFIG.SYSやAUTOEXEC.BATなどの編集に便利な「エディタ」モードの例。このように日本語マルチフォントエディタは、用途に合わせてカスタマイズできます。

❸❶の画面をプリンタで印字した例。対応プリンタも増えました。(カラー印刷は誤差分散により65,536色対応)

❹「パターンエディタ」で作成したデータを、背景に設定できます。

❺バージョンアップした日本語フロントプロセッサASK68K ver.3.0の辞書メンテナンスがウィンドウ上で可能。

❻オリジナルに作成したアイコンパターンの例。

❼アイコンデータや背景データを作成する「パターンエディタ」。文字の貼り付けなど、編集機能も一段とフレンドリーに。

❽512×512ドットの範囲内で65,536色の表示が可能。

❾さまざまなグラフィックフォーマットに対応しています。

❿任意のサイズに縮小・拡大表示可能。

⓫異なる画像フォーマットへのコンバートができます。

⓬「CGAウィンドウ」、65,536色(最大)のコンピュータアニメーション表示が可能です。

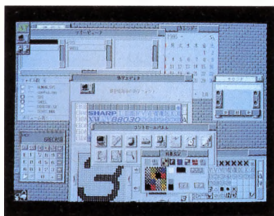
発展性のあるプラットホームとしてのウィンドウシステム、
SX-WINDOW ver.3.0が提供する新たなGUI環境が
さらなるウィンドウ時代を予見する————。
国産オリジナルウィンドウとしての意味、未来への確かなビジョン、
ユーザーインターフェースや高速化へのゆるぎない探求が
ここに凝縮されています。
65,536色表示はもちろん、さまざまな画像フォーマット対応、
イメージデータのコピー&ペースト、
動画、音楽/音声再生をサポートするマルチメディア環境。
そして、何よりもこれらが密接に連携して
統合的にハンドリングできるエキサイティングな環境を創造しています。
未来を照準に入れたウィンドウアーキテクチャ、
そのインテリジェンスがいよいよX68030/X68000シリーズで享受できます。

●65,536色対応、動画ウィンドウ標準装備。

SX-WINDOW ver.3.0 システムキット

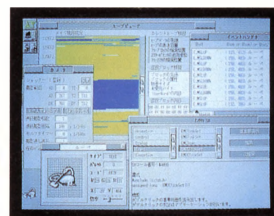
CZ-294SS(5インチ版)/CZ-294SSC(3.5インチ版)各標準価格19,800円(税別)

自然描画に迫る美しい表現が可能な65,536色表示のグラフィックウィンドウを装備。さらにグラフィックウィンドウ内でのアニメーション動画表示、各種グラフィックデータのコンバートも実現しました。またイメージデータの貼り付けなどをサポートした日本語マルチフォントエディタを始め、クリエイティブワークを支援する数々の便利機能を装備、Human68k ver.3.0システムディスクも付属しています。

※メインメモリ4MB以上必要です。SX-WINDOW ver.1.0/1.1/2.0をお持ちの方には有償バージョンアップを行っています。

●「SX-WINDOW開発キット」のサポートツール。

開発キット用ツール集

CZ-289TWD　10月発売予定

SX-WINDOW開発キットをさらに使いやすくするためのツールです。SXコールの簡易リファレンスを簡単に検索するインサイドSX、イベントの発生を常時監視確認するイベントハンドラ、リアルタイムにメモリブロックの利用状況を表示するヒープビューアなど11種のツールが用意されています。 (2MB、ver.2.0)

●SX-WINDOWを楽しく使うためのアクセサリ集。

SX-WINDOW デスクアクセサリ集

CZ-290TWD　標準価格14,800円(税別)

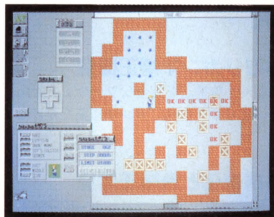
SX-WINDOWをさらに便利に、楽しく使うためのデスクアクセサリ集です。スクリーンセーバ、アドレス帳、電子手帳通信ツール、パズルなど12種類の豊富なアクセサリが収められています。

①キーノート②スクリーンセーバ③スクラップブック④ミュージックボックス⑤ハイパーリンク(電子手帳通信ツール)⑥アドレス⑦スケジューラ⑧ウィンドウアイコニファイ⑨ソフトウェアキーボード⑩パズル⑪ファイルサーチ(ファイル検索ツール)⑫フォントリンカ。 (2MB、ver.3.0)

●SX-WINDOW対応になってさらにパワーアップ。

倉庫番リベンジ SX-68K ユーザー逆襲編

CZ-293AW(5インチ版)CZ-293AWC(3.5インチ版)各標準価格6,800円(税別)

10年にわたるユーザーの投稿など、新作306面が目白押し。まさに倉庫番の最強版がSX-WINDOW上で楽しめます。移動可能先が表示されるAI機能を搭載、またマウスをクリックするだけで簡単に問題を作成できるエディット機能や、キャラクタを替えてちょっと違った雰囲気でゲームが楽しめるキャラクタ変更機能も装備しています。半年で解けたらあなたは天才?です。 (2MB、ver.1.1)

●マルチタスク機能をはじめ、通信環境がさらに充実。

Communication SX-68K

CZ-272CWD　標準価格19,800円(税別)

通信環境をさらに高めたウィンドウ対応の通信ソフトです。マルチタスク機能により他のアプリケーションソフトを実行中でも簡単に通信が可能。また、ホスト局をクリックするだけの自動ログイン機能、初心者にも簡単なプログラム機能、最新モデム(20種類)もフルサポートしています。 (2MB、ver.1.1)

●ウィンドウ対応グラフィックツール。

Easypaint SX-68K

CZ-263GWD　標準価格12,800円(税別)

マウスによる簡単操作、65,536色中16色の多彩な表現、クリエイティブマインドに応えるウィンドウ対応ペイントツールです。同時に複数のウィンドウを開いて編集でき、各ウィンドウ間でのデータ交換もできます。 (2MB、ver.1.1)

●FM音源サウンドエディタ。

SOUND SX-68K

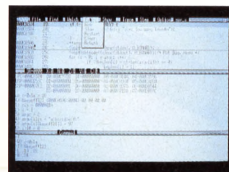
CZ-275MWD　標準価格15,800円(税別)

他のミュージックソフトで演奏中の音色を、簡単に作成、変更できるマルチタスク機能、またエディット、イメージ、ウェーブの3つの編集/確認モードを装備。作成中の音色も50曲の自動演奏でリアルタイムに確認、編集できます。まさにミキサー感覚で音創りが楽しめるツールです。 (2MB、ver.1.1)

●X68030/X68000対応

CZ-295LSD　9月発売予定　NEW

※メインメモリ2MB以上が必要です。
※C compiler PRO-68K/ver.2.0/ver.2.1をお持ちの方には有償バージョンアップを行います。

C compiler PRO-68KのX68030対応版。MPU68030、MC68882の命令セットに対応したアセンブラ、デバッガ、ソースコードデバッガを付属。またHuman68k ver.3.0、ASK68K ver.3.0にも対応。新たにGPIBライブラリ、MC68882対応フローとライブラリを付属しています。

※(2MB,ver.1.1)の表示は、メインメモリ2MB以上、SX-WINDOW ver.1.1以上が必要であることを示します。

※発売予定のソフトの画面は実物とは異なる場合があります。

何がよくなったか…マチエール

Ver. 2.0

プロ仕様ペイントツール　Hyper Image Processor Matier

新機能

1.新ルーペ機能

ノーマルルーペを大幅に機能拡張し、通常のメインメニューの機能をほとんど使用できるようにしました。さらにスーパールーペモードアイコンを新設し、ルーペの状態がノーマルモード／スーパーモードに切り替えることができます。また、ルーペズームアップ・ダウン機能によりルーペ編集中の状態からいきなり倍率を変更することができ、編集作業のスピードを大幅に改善しました。

2.新方式のパレット採用

8色の基本色パレットから好みの色を選ぶと、その色の関係色32色をサブパレットに表示します。

また、サブパレットにもそれぞれ自由に色をセットできますから、全部で264色をパレットとして使うことができます。

3.インクモードの追加

基本的な描画の方法として、インクモードを新設しました。インクモードとは、描画する色を透明インクとみなして、描画先の色が透けて見えるように描くモードのことで、例えば黄色の上に水色で描くと、緑色になります。インクモードには十と一の２種類があり、十は前述の普通の透明インク、一は描画先の色から描画色を色抜きするモードです。インクモードがOFFのときは従来通りの描画モードです。

4.マスク表現の変更

従来の反転表示状態のマスク表示をやめ、パレット切り替えによる色変化表示でマスクを表現します。青プレーンのみ変化させますので、マスクがかかった部分のイメージが比較的よくわかります。

5.SCSI機器への対応

◎ＳＣＳＩスキャナ対応
　SHARP JX-320,JX-325,
　EPSON GT-6000,GT-6500,GT-8000
◎ＳＣＳＩプリンタ対応
　Canon BJC-820C,
　SHARP JX-7000,JX-7010

6.スキャナー読み取りガンマ補正機能

◎スキャナーの読み取り濃度・色補正を完全ソフトウエア対応にしました。各機種の補正機能によらず、自由にガンマ補正をかけることができます。

7.プリンタ出力の高速化と色補正機能

アルゴリズム見直しにより、一部のカラープリンタ（IO-735X, BJC-820C, DeskJet505J）での印刷時間を2倍近く短縮しました。 さらに、印刷色補正機能としてカラープリント時に、ＣＭＹを独立に色補正することができます。その他、印刷位置の上マージン、左マージン、印刷枚数、カラープリンタのモノクロ印刷など もサポートしました。

8.タブレットの絶対座標対応

要望の高かった絶対座標にも対応。画像トレース時などに威力を発揮します。

9.おまけソフトで簡単ステレオグラム

ステレオグラム制作ツール“FLICKER.X”同梱

10.その他

◎スプラッタブラシパラメータの一時保存
◎裏画面ページ内容表示
◎ブラシパターンデータの変更
◎ペースト時の、コピー画像一時表示
◎TIFFフォーマットのグレースケール画像をサポートしました。また、256色パレット付のTIFFファイルや、ビットマップTIFFファイルのロードも可能になりました。

大画面編集も思いのままに

512×512ドット標準画面の解像度では、フィルム出力をして印刷物にするには不足です。

マチエールではメモリー増設により最大2048×2048ドットの画像をリアルタイムに編集できます。

複数の画面で快適編集

512×512 ドットを同時に最大 5画面までもつことができます。（要4Mバイト） 絵のパーツを作っておいたり、２つの画面を合成したり、クリエイティブワークの能率がアップします。画面間の便利な合成機能もいろいろ用意してあります。

立体文字の作成も簡単

どんな図形も簡単に立体表現することができます。

「書体倶楽部」（Zeit 社）のアウトラインフォントや、スキャナでとりこんだロゴマークなども、マチエールで立体文字にすれば、ビジュアル効果も抜群です。

ディザでフルカラーを実現

マッハバンドのない美しいグラデーションは、角度・増減率とも自由に設定可能、4隅の色設定もできます。ぼかし・3次元表現など高度な画像処理も1670万色フルカラーで実現しました。

ジャギーのない高品質

拡大・縮小・変形・バース変形・メッシュ変形など、すべてオーバーサンプリングによるジャギーのない高品質を実現しました。

多彩な編集機能

コピー・クリップコピー・タイルコピー・拡大・縮小・変形・回転・バース変形・メッシュ変形・円筒マッピング・円球マッピング・球面マッピング・領域変換・矩形スクロール・ミラー変形・各種マスク機能など。

専用ソフトなみの画像処理機能

ネガ反転・ディフューズ・ぼかし・モノクロ化・二値化・ランダムノイズ・平滑化・鮮鋭化・輪郭抽出・レリーフ・モザイク・フレア・コントラスト補正・色変換など。

使いやすいスキャナ入力

スキャナ原稿台のプレビュー表示をマウスで範囲指定する簡単操作。

高機能なプリンタ出力

画面の任意の範囲を、 最大A3 までの自由なサイズでプリントアウトできます。

■対応画像フォーマット・入出力機器

画像ファイル形式	PIC・GL3・IMG・RGB・TIFF （Mac・TOWNS互換）
カラープリンタ	Canon BJC-820J（SCSI対応） HP DeskJet 505J SHARP IO-735X・CZ-8PC3・ CZ-8PC4・CZ-8PC5 SHARP JX-7000・JX-7010（SCSI対応） NEC PC-PR406・NEC PC-PR801
モノクロプリンタ	Canon BJ-10V・ESC/P系・PC-PR系
ビデオプリンタ	SHARP CZ-6PV1・NEC PC-VC101
ビデオ取り込み	SHARP CZ-6VT1
カラースキャナ	SHRAP CZ-8NS1・JX-220X （以上純正パラレルボード対応） SHARP JX-320・JX-325（SCSI対応） EPSON GT-1000・GT-3000V・GT-4000 GT-6000・GT-6500・GT-8000（SCSI対応）
タブレット	NS CalComp 33070SER・WACOM SD-510C

※ 機能拡張にともない、メモリー2Mバイトの場合は裏画面が使用できないなど、機能が 一部限定されます。
※ マチエール登録ユーザーの方々には、有償バージョンアップのお知らせをお送りします。

対応機種　X68000/68030（推奨4Mバイト）

価格39,800円（税別）

〒213　川崎市高津区下作延1043　TEL（044）855-4335

microware

OS-9/X68030 V2.4.5

OS-9のX68030対応版、新登場。

68系のリアルタイム・マルチタスク・オペレーティング・システムOS-9に、シャープX68030対応最新版が登場。UNIXライクな操作性と洗練された機能で、X68030の機能を最大限に引き出します。

- ●リアルタイム・マルチタスクOS
- ●マルチウィンドウをサポート
- ●最大10ユーザーのマルチユーザ環境
- ●大容量SCSIハードディスクをサポート
- ●テキストエディタμMACSを標準装備
- ●VJE-γ V2.0による快適な日本語入力

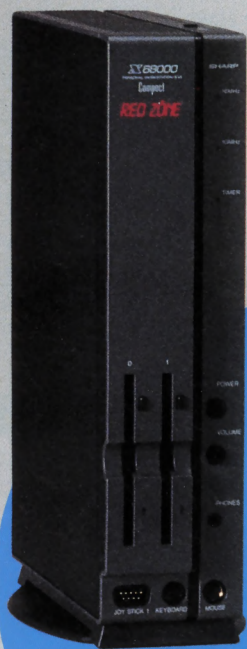
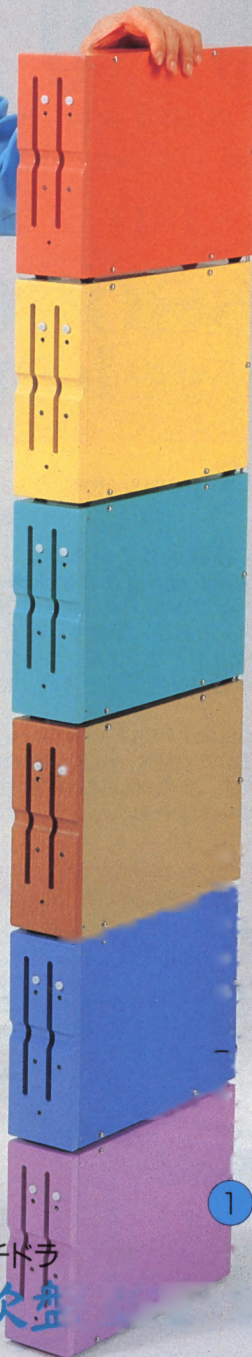
価格
¥25,000(税別)

●OS-9/X68030シリーズラインアップ

製品	価格(税別)
OS-9/X68030 V2.4.5	¥25,000
Ultra C & Professional Pack V1.1	¥45,000
Technical Tool Kit V2.4.5	¥20,000
X Windows V11.5	¥30,000 8月末
MPFMフル動画 V1.0	未定 開発中

※ソフトウェアの内容・仕様は、改良のため予告なく変更する場合があります。

マイクロウェア・システムズ株式会社 〒101 東京都千代田区外神田2丁目17番3号・TEL.(03)3257-9000(代) FAX.(03)3257-9200

EA
COTTON
コットン
い～くぽ～ん
闇に閉ざされた妖精界を舞台に、
お調子者の魔法使いと、しっかり者
の妖精の凸凹コンビが繰り広げる、
痛快、パカポコ ファンタジー シュ
ーティング
BIGでグレートな
©1991,1993 SUCCESS

MK-FD1発売記念セット価格（RED ZONE＋MK-FD1）**¥180,000**(税別)／**¥185,000**（パステルカラー時）

●98用バスマウスアダプタ(MK-MJ1)8月1日発売予定(予価¥3,500税別)

通信販売の方法

★お支払いと商品のお届け方法
●現金書留、郵便振替のいずれかの場合、ご入金確認の後、在庫があれば1週間以内に発送致します。
●代金引換（着払い）にてもお受け致します。
●商品到着後1週間以内の初期不良は新品交換致します。
●現金一括払いのみの取扱いとさせていただきます。
●返品は到着後5日以内に未開封で返送料はお客様負担で、事前に電話連絡をして下さい。
★現金書留、郵便振替の宛先
〒171 豊島区長崎1-28-23Muse西池袋2F パソコンショップ満開
東京 5-568201 パソコンショップ満開
●ご注文の際は、郵便番号・住所・氏名・電話番号を必ずご記入下さい。
★お問い合わせは (03)-3554-7441[月~金 11am~6pm]

（ツクモパソコン本店II 3Fでも展示・販売中！）

パソコンショップ満開

郵便振替 東京 5−568201
口座名 パソコンショップ満開
TEL 03−3554−7441
※住所・FAXは満開製作所と同じ

㈱満開製作所

〒171 東京都豊島区長崎1丁目28-23 Muse西池袋2F
TEL 03-3554-9282
FAX 03-3554-3856

RED ZONEのご購入には、承諾書が必要なのでお電話にてご請求ください。

安いのに親切 TSUKUMO

"情報キャッチステーションツクモ"

ツクモグローバルカード
好評 入会者受付中!
18才以上なら学生でもOK!!
～国内・海外でも使える多機能カード～
ジャックス・VISA、セントラル・マスターのカードです。分割払い、ボーナス払いもOK!クレジット申し込みと同時にカード申し込みOK。
お申し込みは☎03(3251)9898又は各店で
★各店頭では、JCB、日本信販、DC他各種カードも取り扱っております。

SHARP X68000の事なら何でも揃うツクモにおまかせ!
X68000コーナーは、パソコン本店II 3F。ゲームソフト関連は4FにOPEN!!

大好評発売中!!
ツクモ特価販売中!

シリーズ最高峰。ユーザーの期待に応えて更にパワーアップしたX68030!

X68030

- 新たに32ビットCPU(MC68EC030/25MHz)を搭載し、従来機の2.4～4.2倍以上のスピードアップを実現!
- 成熟するウィンドウ環境、使いやすさと高機能を追求し、動画機能、SX-WINDOW Ver3.0搭載
- SX-WINDOWの操作環境を考え、4MBメモリ内蔵
- カラー液晶ディスプレイ接続可能

5インチFDDモデル	CZ-500C-B	定価¥398,000
5インチHDDモデル	CZ-510C-B	定価¥488,000
3.5インチFDDモデル	CZ-300C-B	定価¥388,000
3.5インチHDDモデル	CZ-310C-B	定価¥478,000

★★★★★★★ 超速く ★★★★★★★
X68030用8MB増設RAMボード発売//
●これ1枚でいっきに12MBフル実装
SH-5BE4-8M…ツクモ特価¥46,800

おすすめの組み合わせ

→ハードディスクと…
CZ-500C-B …… ¥398,000
240MBハードディスク …… サービス
ツクモ特価 ¥368,000

→X68000の5インチモデルをお持ちの方には…
CZ-300C-B …… ¥388,000
TS-XFDCA …… ¥9,800
合計定価¥397,800 ツクモ特価 ¥318,000

冬のボーナス一括払(金利手数料なし)詳しくはお問い合わせ下さい。

目のつけどころがツクモでしょう。

●X68000 & X68030シリーズ対応3.5インチフロッピーディスクドライブ

TS-3XR シリーズ
〈仕様〉
- 3.5インチ2DD/2HD/2HCフォーマット対応
- ユーティリティソフト付属(デバイスドライバー/フォーマッター)
- 標準サイズケーブル付

TS-3XR1(1ドライブ) 定価¥44,800 …… ツクモ特価 ¥34,800
TS-3XR2(2ドライブ) 定価¥57,800 …… ツクモ特価 ¥45,800
●Compact XVI/X68030シリーズでお使いの方は、別売ケーブル(TS-XR5CA特価¥6,800)が必要です。

●X68000 Compact & 68030シリーズ対応フロッピーディスクドライブ

TS-5XR シリーズ
〈仕様〉
- 5インチ2HD/2DDフォーマット対応
- ドライブ番号切り換スイッチ付
- Compact XVI/X68030用ケーブル付

TS-5XR1(1ドライブ) 定価¥53,800 …… ツクモ特価 ¥35,800
TS-5XR2(2ドライブ) 定価¥72,800 …… ツクモ特価 ¥47,800

X68000/030シリーズ用RAMボード

	ツクモ特価
SH-6BE1-1ME(CZ-600C専用)	¥11,000
PIO-6BE1-AE(ACE/PRO/PRO2シリーズ用)	¥11,000
PIO-6BE2-2ME(拡張スロット用)	¥23,000
PIO-6BE4-4ME(拡張スロット用)	¥39,000
SH-5BE4-8M(X68030シリーズ用)	¥46,800
CZ-6BE2A(XVI専用)	¥42,500
CZ-6BE2D(Compact XVI専用)	¥39,000
TS-6BE2B(CZ-6BE2A/D用拡張RAM)	¥29,800

大容量記憶装置

MOが今 一番トレンディ
Logitec 3.5インチ光磁気ディスクユニットセット
LMO-FMX330-TS …… ¥178,000
SCSIケーブル …… サービス
ツクモ特価 ¥128,000
※MOメディア、レンズクリーナー、フィルター付属
※Human68K Ver3.0以上が必要です。

おすすめ SCSIハードディスク
- 120MBハードディスク ツクモ特価 ¥48,000
- 200MBハードディスク ツクモ特価 ¥59,000
- 240MBハードディスク ツクモ特価 ¥68,000

コンピュータアート

◆スーパーグラフィックツールセット
その1. 慣れてしまうとマウスがいらない
NS Calcomp製 Drawing Pad(タブレットセット) …… ¥76,500
サンワード Matier(マチエール) …… ¥39,800
ツクモ特価 ¥95,000 合計定価¥116,300

その2. ハイクオリティなのにこんなに安い
ヒューレットパッカード HP Desk Jet 505J(インクジェット) ¥99,800
ヒューレットパッカード カラーキット …… ¥12,000
アーベル プリンタケーブル …… ¥4,800
サンワード Matier(マチエール) …… ¥39,800
ツクモ特価 ¥112,000 合計定価¥156,400

◆プリンター
48ドットカラー熱転写プリンター ツクモ特価
CZ-8PC5-BK …… ¥39,800
カラーイメージジェット ツクモ特価
IO-735X-B …… ¥130,000

◆カラーイメージスキャナー
CZ-8NS1 … ツクモ特価 ¥99,800

パソコン通信

★モデム
オムロン MD144XT10V 定価¥52,800 ツクモ特価 ¥44,800
マイクロコア MC14400FX 定価¥46,800 ツクモ特価 ¥39,800

★通信ソフト
- たーみのる2 ツクモ特価 ¥13,000
- Communication SX-68K ツクモ特価 ¥16,800

MIDIコンピュータミュージック特選セット

特選Aセット
◆SC-55MKII ¥69,000
◆SX-68MII ¥19,800
◆Mu-1 Super ¥25,000
(キャンペーン版)
合計定価¥113,800 ツクモ特価 ¥90,000

特選Bセット
◆CM-500 ¥115,000
◆SX-68MII ¥19,800
◆Mu-1 Super ¥25,000
(キャンペーン版)
合計定価¥159,800 ツクモ特価 ¥132,000

通信販売のご注文は下記フリーダイヤルへ。
全国どこからでも通話料無料
受・注・専・用 フリーダイヤル 0120-377-999
通販センター 03-3251-9911 商品についてのお問い合わせは各店又は通販へ。

クレジット払い
月々¥3,000以上の均等払いも頭金なし、夏・冬ボーナス2回払いも受付中!

カード払い(¥5,000以上)
通信販売での御利用カード、ツクモグローバルカード、VIPカード、セントラル、ジャックス※御本人様より電話で通販部へお申し込み下さい。

各種リース払い
くわしくは各店にお問い合わせ下さい。ケースに合わせてご相談にのります!

全国代金引き換え配達
お申し込みは☎03-3251-9911へお電話1本!
配達日の指定もできます。

現金書留払い
〒101-91 東京都千代田区神田郵便局私書箱135号
ツクモ通販センター Oh!X係

銀行振込払い
事前にお届け先をご連絡下さい。
三和銀行 秋葉原支店(普)1009939 ツクモデンキ

秋葉原各店
営平日AM10:45～PM7:30
日・祝AM10:15～PM7:00

パソコン本店/荒井

ツクモパソコン本店II 3F
☎03-3253-1899(直通)(担当/荒井)
ツクモパソコン本店II代表☎03-3253-4199 休毎週木曜日
ツクモニューセンター店 ☎03-3251-0987(担当/沢栄)休毎週木曜日
(下取り交換、中古販売も行っております。)
※定休日が祝日と重なる場合は営業致します。

名古屋各店
名古屋1号店 ☎052-263-1655
営AM10:00～PM7:00 休毎週火曜日
名古屋2号店 ☎052-251-3399(担当/松原)
営AM10:00～PM7:00 休毎週水曜日

札幌各店
ツクモ札幌店 ☎011-241-2299(担当/田口)
営AM10:30～PM7:30 休毎週木曜日
DEPOツクモ2番街店 ☎011-242-3199(担当/鈴木)
営平日AM10:40～PM7:30
日・祝日AM10:10～PM7:00 休毎週木曜日

ツクモは「スーパーX PRO SHOP」です。
PRO STAFF ツクモ
九十九電機株
〒101-91 東京都千代田区神田郵便局私書箱135号
★商品のご注文は在庫確認の上お願いします。★表示価格には消費税は含まれておりません。

Mac
ユーザーの皆さん、
「マックユーザー」
の時間ですよー。

Oh!X

Graphic Gallery

DōGA CGアニメーション講座

今月は，連載のほうとは直接関係ありませんが，最近のCG作品のなかからいくつか紹介しましょう。

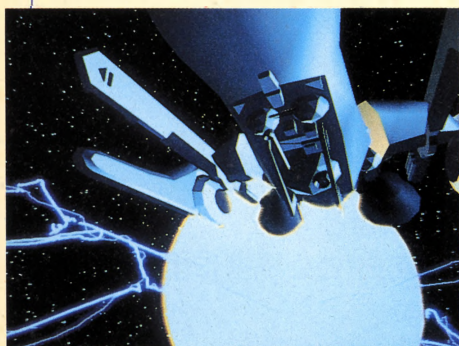
1．マッピング用画像データ

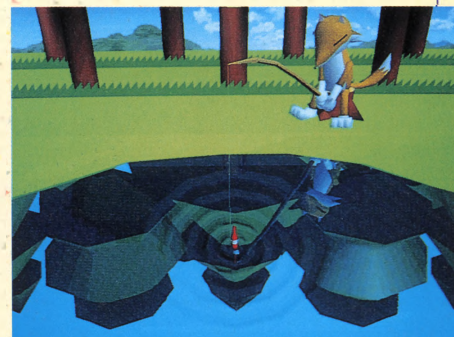
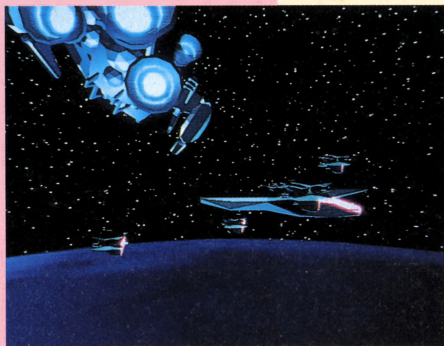
2．木星とガニメデ，そして，遠くにイオを配置してみた

1は，木星のマッピング用画像データです。青森県の佐々木さんからの投稿データで，ほかにも地球，月，木星とその衛星のマッピング用画像データを投稿してくださいました。

2，3はともに，そのマッピングデータをTAMEN.Xで発生させた球に張り付けさせただけのものです。

これらのリアルなマッピング用画像は，NASAのデータを流用したものでもなく，また，木星まで行ってスキャナにかけたものでもありません。単なる手描き！（著作権は大丈夫だ）もう天才的！　ただ，あまりにデータ容量が大きいので，配付の方法については，検討中です。

3．CGAマガジン3号にあった地球のマッピングデータに手を加え，雲を描いたもの

EPA2.Xで有名な「宇宙人」森山さん制作のデモの一部。このデモでは，全編いたるところに「森山効果」（ピカピカ炸裂）が使われています。

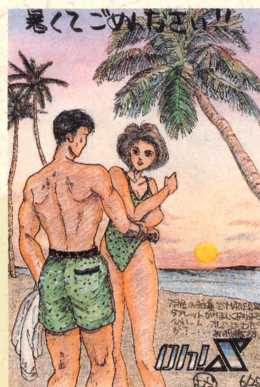
なお，左側の写真の曲面部分にまったくマッハバンドがないのは，最近導入された誤差拡散法を組み込んだREND.Xの実力です。

126ページ「ホームズに挑戦」のコーナーで紹介した河内さんの投稿です。7月号の連載を応用して，映り込みが揺れながら，波紋が広がっていくアニメーションを制作されました。ちょっと苦しいけど，なかなか面白い効果が出ています。このように連載を応用してくれると，書いているほうとしてはとても嬉しいです。

暑中見舞いだ！
Oh!X reader'sぎゃらりぃ

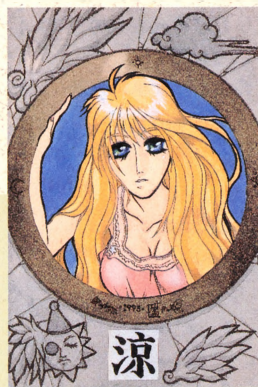
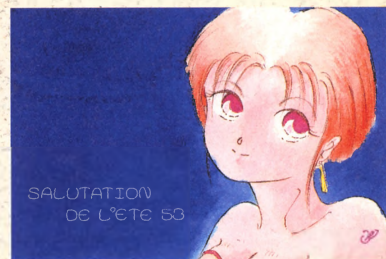
今年は，夏らしい天気があまり見られませんでした。うむ，残念。代わりに，読者からの熱いカラーイラストで，過ぎ去った夏を思い出してください。

暑中おみまい申上げます

▶日高　光代（宮崎県）

▲姉帯　寛（茨城県）

▲玉野　健一（奈良県）

▲大村　正行（北海道）

▲岩瀬　貴代美（福岡県）

▲川崎　光臣（茨城県）

▲岡田　徹（静岡県）

▲青木　一師（奈良県）

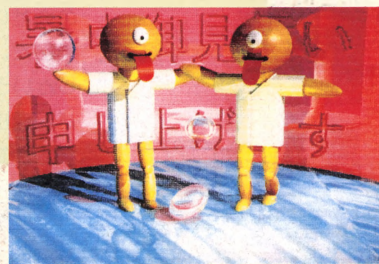
▲徳物　信生（Miami）

▶北本　信幸（石川県）

▶藤原　彰人（岡山県）

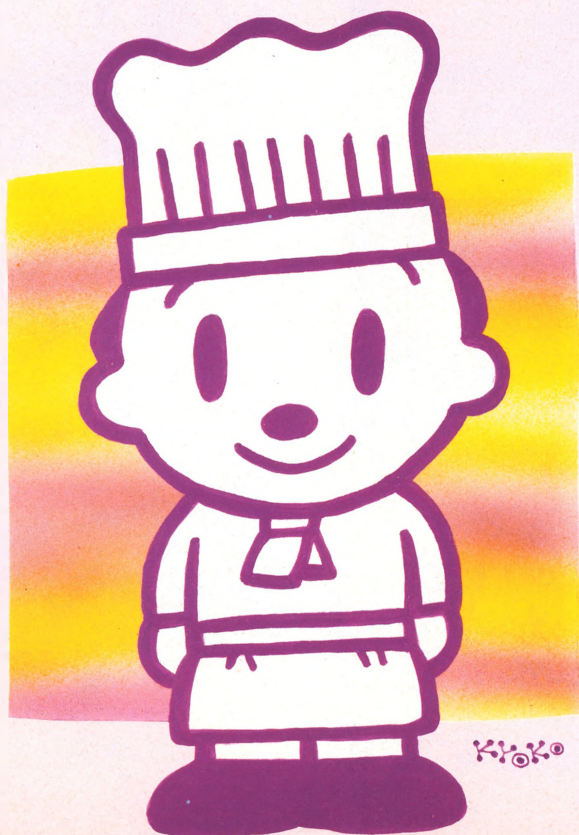
▶鈴木　貴久（神奈川県）

［第29回］
注文どおりの料理店

江口 響子

響子inCGわ〜るど

大学時代の同級生Mから手紙がきた。

レストランを始めました
どなたも，けっしてご遠慮はいりません
RESTAURANT
西洋料理店
CITYCAT HOUSE
都会猫軒

・あなたが召し上がりたいと思われるものをこちらで想像してお作りいたします
・お好きな食べ物，最近の体調，仕事の近況など同封の用紙にご記入のうえ返送ください
・メニューは秘密，日時はのちほどご連絡いたします。

あいつ，独立したのか。そういえば，学校に弁当を彼女の分までこまめに作ってきていたな。よし，行ってみるか。私は返事を書いた。すぐにMから連絡があった。

日曜日の昼下がり。店はオフィス街のなかにひっそりと建っていた。扉を開けると，白い帽子をかぶったシェフ姿でMが出てきた。やあしばらく，卒業以来だねとひとしきり挨拶を交わしたあと，「今日の客は君ひとりなんだ。特別メニューを用意したよ。気に入ってくれるといいけれど」と言った。

案内された部屋には大きなガラス窓があって，秋の柔らかな光がテーブル一杯に注いでいる。Mおすすめのよく冷えた白ワインに口をつけると，すぐに料理が運ばれてきた。温かいコーン・ポタージュ，生牡蠣レモン添え。どれも私の好物である。続いて出てきた大皿に，私はぎょっとした。それは，「お助けガール」の頭だった。

「お助けガール」。一般事務補助マシン。電話の応対，経理業務，ワープロ打ちなどをする秘書代行ロボットというのがわかりやすいかな。私が現在の会社に入ってから，開発をずっと手掛けてきた製品だ。が，CMはオンエアされ，いよいよ売り出しというところで，急に発売中止となってしまった。理由はこうだ。

人件費削減のための機械化が進み，企業は人をあまり採用しなくなっていた。ひっくりかえせば仕事にありつけない人間が増えてしまったのだ。事態を重くみた政府は，雇用促進のためマシンより人間を使うようにと，政府広報でキャンペーンを繰り広げた。人間の能力を尊重しましょう……と。人を採らない企業は非人間的だと思われて，イメージがなんとなく悪くなっていった。失われた信用を回復しようと，企業はこぞって機械導入をやめ，人を使い始めた。人間にとって代わるような機械「お助けガール」なんてもってのほか，というわけだ。

「お助けガール」開発チームは解散し，私は営業に回された。もうエンジニアとしてのピークはすでに過ぎていた。いままでがんばってきた十数年はなんだったのだろう。倉庫に眠る「お助けガール」はいつ処分されるかわからない。手元に残したかったので１体買い取ろうとしたが，企業秘密を守るためそれはできないと会社に断られた。そして，これらのことをMへの返事に書いたのだった。

主菜は「お助けガール」そっくりだったが，もちろん本物ではない。味は白身魚のようでもあり，柔らかい牛のヒレステーキにも似ていた。量は多かったが，あっさりとしていたのですっかり平らげてしまった。デザートは，チーズケーキ。ていねいに入れた濃いコーヒーが出て，これが私のために用意してくれたMの特別メニューをしめくくった。

帰りぎわ，Mは片目をちょっとつぶってこう言った。

「物を作るって楽しいね。作った物が人に喜んでもらえたときは，特にね」

私は笑って，おいしかったまたくるよと手をふった。次に彼はどんな料理を作ってくれるのだろう。

ビデオゲームアンソロジーシリーズ第6弾は，Oh!Xスタッフのなかでも移植希望の声があがっていた，あの「ぶたさん」です。某氏の祈りが届いたのでしょうか。「叩けよ，さらば開かれん」ってとこかな。

ぶたさん

ゲームセンターのゲームは，パズル以外の固定画面ゲームに記憶に残るモノが多いといわれている。そのなかでもかなり通好みのヤツが，ぶたのぶたによるぶたのためのゲーム「ぶたさん」である。この一見地味なゲームが，多くのファンの爆発的な要望に応えるかのように，電波新聞社のビデオゲームアンソロジーシリーズの最新作として，リリースされることになった。オリジナル版のデビューは1987年の年末だが，ぶた尽くしと呼びたくなるほどの独特の世界は，少しも色あせてはいない。ぶたをあやつり爆弾を投げ，敵のぶたを退治するだけという単純なルールに，ほのぼのとしたグラフィック，さらにそれらに隠された高度なプレイに追従する緻密なゲームシステム。それらがほどよく，まるで美味しいトンカツの肉と衣のようなハーモニーを形成しているデリシャスでスペシャルなゲームが，この「ぶたさん」なのである。

いまからでも遅くない。理性を捨て，ぶたとなって爆発するのだ！　(八)

X68000用　5″2HD版　5,900円(税別)
電波新聞社　☎03(3445)6111

新作への期待大で順位変動

	タイトル	(前回順位)
1.	ネメシス'90改	—
2.	コットン	2
3.	クレイジークライマー/クレイジークライマー2	3
4.	スタークルーザーII	—
5.	ストリートファイターII	4
6.	スーパーリアル麻雀PII＆PIII	—
7.	EG Word	8
8.	SX-WINDOW開発キット	5
9.	C COMPILER PRO-68K NEW KIT	—
10.	Easydraw SX-68K	9

9月号の読者アンケートハガキのなかの「期待している新作ソフト」を集計したものです。上位10点のなかに，新登場のものが今月は4タイトルもあります。この順位の変動は，コナミの「悪魔城ドラキュラ」をはじめとして，大きな注目を集めていた新作ソフトが次々に店頭に並び始めたことと，秋以降に発売されるソフトの情報が出てきたことによるものでしょう。

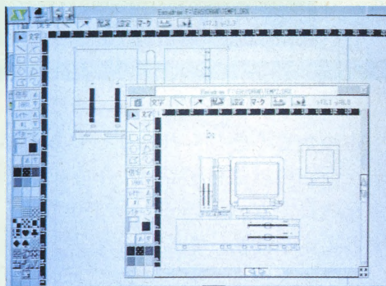
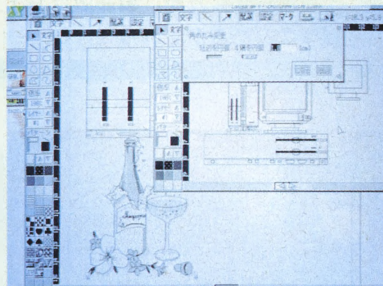
噂の「ネメシス'90改」は，完成のめどがたったとの発表で，いきなり1位に躍り出てきました。同様に，4位の「スタークルーザーII」と6位の「スーパーリアル麻雀PII＆PIII」も，先月号での紹介直後にベスト10入りです。どれもまだ発売日などは決定していませんが，開発は順調に進行している模様です。今月はこれらについての新しい情報はありませんが，来月号ではもう少し詳しくお伝えできるかもしれません。どのタイトルもこれまでに発売希望の声が高かったものばかりなので，ファンにはうれしい限りでしょう。待ち遠しいですね。

そして，やはり強い「コットン」と「クレイジークライマー/クレイジークライマー2」は先月号と順位変わらずで，人気のほどがうかがわれます。

今回の集計では，ゲーム以外のソフトはちょっとおとなしいようですが，これはあくまでも順位のうえでの話。それぞれのソフトに対する期待度は高まっています。シャープからもツールや開発キットなどのソフトが順次，発売される予定になっていますので，今後の順位変動は予断を許しません。ゲームソフトで盛り上がるか，SX関係はどうなるか。

さあ，来月号は？

Easydraw SX-68K

ドローツール「Easydraw SX-68K」が発売された。8月号で述べた問題点は製品版ではほとんどが改善されている。スムージング処理やベジェ曲線もサポートしているほか，レーザープリンタドライバも付属。動作にはSX-WINDOW ver.3.0およびメインメモリ4Mバイト以上が必要だ。

X68000用　3.5/5″2HD版　19,800円(税別)
シャープ　☎03(3260)1161

宝魔ハンターライム3

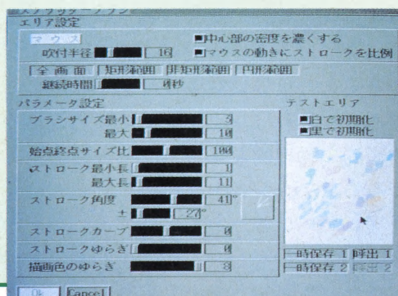
連続モノとして1話ずつお手頃価格で発売される，この「宝魔ハンターライム」シリーズ。キャラクターデザインは「うる星やつら」「らんま1/2」などを手がけた中嶋敦子氏で，ビジュアルの可愛さは文句なしだ。

今回の舞台は病院。となると，期待のライムの変身シーンは当然ながら「白衣の天使」。サブキャラも増えて，今後の展開も楽しみだね。

X68000用　3.5/5″2HD版　1,500円(税込)
ブラザー工業(TAKERU)　☎052(824)2493

画面はPC-9801版です

MATIER ver.2.0

人気のペイントツール「MATIER」もいよいよver.2.0の登場だ。ルーペ機能の拡張やパレットの改良などのほか，タブレットの絶対座標対応，スキャナ読み取りガンマ補正機能，スキャナやプリンタへのSCSI対応などの新機能の追加で，描画環境はさらに強化される。

発売は10月頃の予定で，現バージョンの登録ユーザーは有償バージョンアップができる。

X68000用　5″2HD版　39,800円(税別)
サンワード　☎044(855)4335

発売中のソフト

★Easydraw SX-68K　シャープ
X68000用　3.5+5″2HD版　19,800円(税別)

★クレイジークライマー/クレイジークライマー2　電波新聞社　8/27
X68000用　5″2HD版　5,900円(税別)

★宝魔ハンターライム2　ブラザー工業(TAKERU)
X68000用　3.5/5″2HD版　1,500円(税込)

★レッスルエンジェルス　ブラザー工業(TAKERU)　8/末
X68000用　3.5/5″2HD版　4,900円(税込)

新作情報

★Ultra C & Professional Pack V1.1　マイクロウェアシステムズ
X68030用　3.5+5″2HD版　45,000円(税別)

★Technical Tool Kit V2.4.5　マイクロウェアシステムズ
X68030用　3.5+5″2HD版　20,000円(税別)

★宝魔ハンターライム3　ブラザー工業(TAKERU)
X68000用　3.5/5″2HD版　1,500円(税込)

★コットン　EAビクター　9/24
X68000用　5″2HD版　9,800円

★SX-WINDOW 開発キットWorkroom SX-68K　シャープ　9/末
X680x0用　3.5/5″2HD版　価格未定

★C COMPILER PRO-68K NEW KIT　シャープ　9/末
X680x0用　3.5/5″2HD版　価格未定

★スーパーリアル麻雀PⅡ&PⅢ　ビング　10/中
X68000用　5″2HD版　12,800円(税別)

★MATIER Ver.2.0　サンワード　10/末
X68000用　5″2HD版　39,800円(税別)

★項劉記　光栄　10/29
X68000用　5″2HD版　12,800円(税別)

★ぶたさん　電波新聞社　10/下
X68000用　5″2HD版　5,900円(税別)

★SX-WINDOW 開発キット用サポートツール集　シャープ　10/末
X68000用　3.5/5″2HD版　価格未定

★ネメシス'90改　SPS
X68000用　5″2HD版　価格未定

★マージャンクエスト　SPS
X68000用　5″2HD版　価格未定

★ギャラクシーシェイカーズ　ブラザー工業(TAKERU)
X68000用　価格未定

★ロボスポーツ　イマジニア
X68000用　5″2HD版　価格未定

★Traum　象スタジオ
X68000用　5″2HD版　価格未定

★鮫！ 鮫！ 鮫！　KANEKO
X68000用　5″2HD版　価格未定

★達人　KANEKO
X68000用　5″2HD版　価格未定

★エアバスター　KANEKO
X68000用　5″2HD版　価格未定

★サバッシュⅡ　ポプコムソフト/グローディア
X68000用　5″2HD版　価格未定

★麻雀悟空・天竺への道　シャノアール
X68000用　5″2HD版　9,800円(税別)

★スタークルーザーⅡ　アルシスソフトウェア
X68000用　5″2HD版　価格未定

★餓狼伝説2　魔法株式会社
X68000用　5″2HD版　価格未定

艶姿いくぽ〜ん娘の旅

WILLOW大好き娘の「コットン」も，いよいよ発売目前。もうすぐみんなのおウチにやってくる。先月号の評価版では「100%完全移植」とお知らせしたが，製品版はちょっと違うようだ。さて，どうなってるか。れっつ　GO〜！

Yaegaki Nachi
八重垣 那智

先月の記事で，最近横スクロールシューティングがないことを嘆いていたが，気がつくとX680x0シリーズには，今月も紹介するこの「コットン」，待望の「ネメシス'90改」と期待作が並んでいるではないか。とりあえずこういうゲームがやりたいときには，X680x0シリーズが世界でいちばん向いていることを確認できたところで，今月もコットンと妖精界の旅に出ることにしよう。くれぐれも連射スティックを忘れないように。おやつは300円までだ。

もういちどコットン

先月，おおまかなゲームルールを紹介したこの「コットン」であるが，軽くおさらいしておこう。ゲームの基本は，レバーで避けて，2つのショットボタンで正面と下を攻撃する横スクロールシューティングゲームである。敵を倒してパワーアップしたり，アイテムを溜め射ちして，魔法攻撃もできるようになっている。そして，面の最後のボスを倒すと，かわいくて楽しいストーリーデモが見られるようになっている。このストーリーの楽しさと，ゲーム自体の真面目なところの組み合わせが，世にも不思議な「コットン」の世界を作り出しているといっていいだろう。

その，気になるストーリーは，世界を闇の波動によって，永遠に夜のままにしてしまった悪い奴を倒すというものなのだが，

X68000用　5"2HD版2枚組　9,800円(税別)
EAビクター　☎03(5410)3111

普通の1面……，じゃなくて誘導モードの1面!?

実際には，脳天気な主人公のコットンは，ワガママな自分の欲望のために戦っているだけなのである。自分の大好きな「WILLOW」を食べたいためだけで身を危険にさらし，しかし，結果的には世界のために戦ってしまうのである。こういった，典型的なヒーローやヒロインとは違う，妙なキャラクターの存在感も，このストーリーや，ひいてはゲームまでをも引きたてている。これが「コットン」の最大の特徴だ。

スゴイぞコットン

現時点の状況を整理してみると，先月書いたようにアーケードゲームそのままの100%移植ではなく，オリジナルの欠点を補った120%の拡張移植という，あまり例をみないことになっているようだ。このプラスαは，部外者や移植メーカーの独自の勝手な解釈ではなく，より素晴らしい「コットン」がプレイできるようにするための，オリジナルメーカーによる移植ならではの拡張や改良によるものだ。

オリジナルがゲームセンターで短命だった事実からある程度わかるように，「コットン」は歴史に残る名作というわけではないし，もちろん大ヒットしたわけでもない。丁寧な作り方や，演出のインパクトで，比較的オーソドックスな横スクロールシューティングを「化けさせた」という印象が強い。そういった面では，「コットン」の演出的要素やゲーム構成はいま現在では，やや古めかしく感じられることもあるだろう。

今回の拡張移植では，そういった問題点を改良するべく，巧みにX680x0オリジナルの要素や技術，さらにはオリジナルの敵キャラクターまで取り入れ，より完成度の高い「コットン」をプレイすることができるようになっているのである。しかし，ゲームの基本コンセプトは踏襲されており，あくまでも「コットン」という枠からははみだしていない。

移植というと，普通はオリジナルを目標として，どこまで似せるか・似ているかという点に絞られて比較されているが，「コットン」ではそういう次元を超え，オリジナルを超えた世界に突入してしまっているのである。それは，小手先の話題のためや，技術的・物理的なギャップを埋めるための，よくあるアレンジとは，決定的にその性格を異にしている。ある意味では，理想の移植といえるかもしれない。オリジナルに似ていないことが，必ずしも悪いことではないという，よい見本になるであろう。

遊ぼうコットン

では，ここで各面の紹介をしておこう。いつものひと口紹介風に，ファンタジーの世界でどんな戦いが繰り広げられるのか，チェックしてみよう。

1面：街

最初は水辺からスタートする。とにかく敵を逃さず撃ち落としてパワーアップに専

千歳飴みたいなレーザー。もちろん当たると死ぬ

念しよう。岩がぐるぐる回る中ボスは，連射が遅いなら妖精を飛ばしたり，魔法を使って早く倒すほうがよいだろう。魔法はボタンを押したままの溜めが必要なので，あらかじめ使うと決めたら惜しまず，きっちり使っていくほうがよい。ボスはひたすら避けて撃つだけなので，これはとにかく集中力だけの問題。

最後のお茶のボーナスステージは，もちろん隠れボーナス狙いで，1個も取らないのが基本である。これについては，どの面でも同じ。

オリジナルには安全地帯があったのだが……

炎で燃え尽き，フッ飛ばされるコットン。合掌

2面：森

敵が弾を容赦なく撃つので，急に難しくなった印象を受ける面。特に音符を8方向に飛ばす敵などは，確実に仕留めないと苦労する。中ボスの巨木では，飛んでくる根っこを壊したときの水晶石がショットにひっかかって，攻撃力が下がらないようにするといい。

3面：岩場からお城

この面から，地形もだんだん狭くなっていく。ミスしたあとたて続けにズルズルとミスするということのないように。この面ではボスがかなりの強敵。左下隅は剣の攻撃からは安全地帯なので，危ないときにはそこでやり過ごすといいかもしれない。さらに，シマシマのヘビが出てくる卵や誘導弾は必ず壊すように。

4面：神殿

水が流れてキレイなのはさておき，狭いので苦戦は必至。特に柱を壊したりしながら進むので，敵が比較的自由になることが多く，弾がボロボロ飛んでくるので，気を抜かないようにしよう。

5面：地下洞窟

とにかく狭くて，スクロールも複雑なのが，そもそも強敵。中ボスのカニの泡は，無害だが移動を妨げるので，うまく片方ずつ倒したい。オリジナルとは別ものの目玉の怪獣のボスは，画面右端からジャンプするように見せかけて，そのまま突っ込んでくるフェイントにだけ注意。

6面：火山

飛んでいるドラゴンは，距離を空ければそれほど怖くない。地形から火を吹き出すところは，確実に壊すようにしないと，大変なことになる。中ボスの牛は，わざわざ挟まれると上から落ちてくる岩にやられるので，間違っても実験しないこと。

7面：最終面

敵の親玉との最終決戦。ここでも水晶石がショットをふさいでしまい，敵にショットが当たらなくなる状況が起こる。一瞬撃つのをやめるか，思い切って回収したほうがいいだろう。

おいしいコットン

X680x0版の「コットン」では，いくつかの特殊機能が内蔵されている。どれもこれも，ユーザーサイドに立ったもので，「コットン」をより楽しくプレイすることができるようになっている。いくつか，その機能を紹介してみよう。

まずはトレースプレイ。自分のプレイを保存することができる。ストーリーデモ中の処理まで保存されているのは，ちょっと笑える。

次は設定の保存機能。通常のゲームでは，キー設定とか難易度設定ができても，ゲームを終了するとキレイさっぱり消えてしまっていたが，「コットン」では，その設定を自動的にセーブするようにできる。普通のゲームのように，セーブを無効化することもできるので，うっとうしいという人は切っておけばよいだろう。

ちなみに，難易度設定で「YOU DO！」というのがあるのだが，これは敵の弾すべてが誘導弾になる，シャレ(シャレでは済まない気もするが)のモードのようである。もちろんこれは，オリジナル版にはついていない。

ほかにも起動時一気読み(4Mバイト対応)やデータのハードディスクへのインストールもサポートされているので，遊ぶにあたって工夫すれば，いろいろ楽しめるようになっている。なかなかのサービス精神ではないだろうか。

やっぱりコットン

最初に，拡張移植を手放しで褒めたが，やはり，オリジナルの「コットン」が好きでたまらない人には何かと抵抗があるかもしれない。しかし自機のシステムなどの，プレイヤーに近い部分がきっちり移植されているので，違和感を感じるまでには至らなかった。

ここで，2周目がないとか文句をつけてもいいのだが，せっかくの意欲作にそういったケチはつけないようにする。X680x0ならではの「コットン」が，移植の新境地を開くことに期待しよう。

集めているようだが，実は必死によけているのだ

コットンだぼ〜ん

この「コットン」の移植では，ポーズ時の音声合成などが追加されており，サウンドもそれなりに拡張されているようです。特にポーズのときの「休んでぽん」というのはかなり絶妙で，気に入ってしまいました。こういう枝葉の部分で楽しませてくれる余裕が作る側にあるということは，ゲーム本体の出来具合に自信があるということにつながると考えられるわけです。余裕ついでに，パッケージにWILLOWが入っていると気分もばっちりなんだけど，発売直後に買わないと腐っていそうだから，やっぱりいいや，うんうん。

総合評価 (0〜10)	
ゲーム性	★★★★★★★★
技術	★★★★★★★★
サウンド	★★★★★★★★
グラフィック	★★★★★★★★
お買い得度	★★★★★★★★
ビッグでグレート	★★★★★★★★★

X68000を楽しむ5つの方法

Nishikawa Zenji
西川 善司

ユーザーが自らの手で自分のやりたいことを実現する。その意味で，X68000は真の「パーソナル」コンピュータだといえるだろう。自分で作る，という熱い思いが寄せられた昨年のX68000芸術祭の作品をみてみよう。

ラストX68000ヒーローこと山下章氏がプロデュースした「第1回全日本X68000芸術祭」。Oh!Xでもおなじみの文月涼氏の「TORNADO」がグランプリを獲得して幕を閉じたわけだが，レベルの高い個性的な作品が多く寄せられ，大成功を収めたことは記憶に新しい(最後の本選が行われたのが1992年4月12日，もう1年以上も前のことなのに)。第2回がどうなるかは今後のX68000シリーズの売れ行きしだいといったところだが，まずは第1回の作品の優秀なものをユーザーにお届けしようと，電波新聞社がひと肌脱いで，今回の「The World of X68000」発売の運びとなった。

この「The World of X68000」は，いわゆるパッケージソフトではなくて書籍の形態をとっている。全64ページの冊子1冊と5インチフロッピーが3枚という構成。収録されているのは5つのゲーム作品だ。音楽作品や一発芸ソフト，ツールなどの作品も収録してほしかったとも思うのだが，そのあたりは少し残念。

冊子には，収録ゲームの遊び方と制作者のコメントなどが書かれている。このコメントは読む人によっては意外と貴重な情報かもしれない。ゲームを制作時に使用したツール一覧から，技術的な話，プログラミングの際に苦労した点，BGM楽譜，あげくの果てには現在のゲーム業界に対する危惧，

X68000用　5"2HD版3枚組4,800円(税込)
電波新聞社　☎03(3445)6111

お姫さまはなんでも投げちゃうのだ

将来ゲームクリエイターになる人へのアドバイスまで書かれている。お小遣いの少ない君は「んー，そろそろぼくも将来のことを考えたいのでこれを買ってほしいなり～，マミー&ダディ」とおねだりすれば，おそらく買ってもらえるだろう(「ゲームデザイナーなんてそんなヤクザな商売，ママは許しませんよ」といわれたなら旅に出ろ！)。そのほか，巻末には市販ゲームソフトの隠れコマンドやマル秘テクニックなども載っており，ゲーマー必見の情報が盛りだくさんだ。行き詰まってしまったあのゲーム，このゲーム，みんなエンディングが見られるかもしれないぞ！

ああっ！お姫さまっ！

高倉正充氏作

主人公は何でも持ち上げることのできる怪力お姫さま。メイドたちが運んでくるさまざまな家財道具を，城壁をよじ登る変態サル目がけて投げ落として撃退するという単純明快，昔流行ったゲームアンドウォッチ系のゲーム。

サルは3種類いて茶色，白色，灰色の順に強い。そして家財道具はその重さによってサルに与えるダメージが違う。威力は実際にそれを投げてみて確認するしかないが，最新GUI機能により，見た目で重そうなものは威力があるようになっている(例外もあり)。重いものや長いものを一気に複数のサルに当てて突き落とすと高得点。特殊アイテムにムチがあって，これを取るとお姫さまは女王様に変身し，家財道具をいちいち落とさずに直接ムチでサルを叩き落とすことができる。さらにジョイスティック操作でお姫さまの嘲笑を聞くことも……。

最初は少し単調なゲーム展開だが，ステージが進むとサルの動きもトリッキーになってきて，スリル満点ハラハラドキドキ。ルールが単純なので小学校低学年のお子さんがいても家族そろって楽しめるぞ。「マサオ，なんだ，もうおしまいか」「父さんだって1面もクリアできなかったじゃないか」「そうよ，あなた」「こりゃマサオに一本取られたな，どわっはっは」といったなごやかな団欒がこのゲームを媒介に生まれることをこの私が保証しよう。

FORTRESS ATTACK

柴原章宏氏作

「ゲームはシューティングだぜ！」という

Formula X

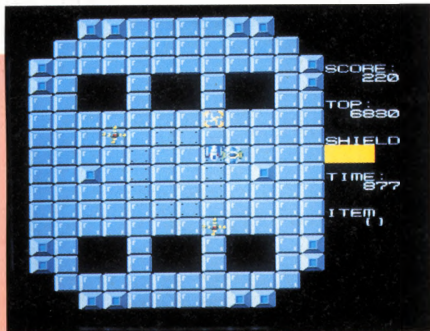
多人数ゲームだが，ひとりでも遊べる

GJは時間との戦いだ

熱いお兄さんが最近は少ないそうだが，そんな冷め果てたハートを揺さぶりそうなのが，この「FORTRESS ATTACK」だ。

ゲームはグラディウスタイプの横スクロールシューティング。敵を破壊しつつ進み，最後に控えているボスを倒せばステージクリア。オーソドックスなゲーム展開だ。では，どのあたりがシェイキン・マイ・ハートかというと，それは個性的な「自機」なのだ。

自機に装着されるオプション兵器の機能の仕方が微妙に違うタイプ1とタイプ2それに，オプション兵器はまったくなしで自機のショットがどんどんパワーアップしていくタイプ3の，3種類のなかから自機をセレクトしてプレイできるのだ。またオプション装着時にトリガBを押すと弾道を変化させられるので，状況や敵の種類に適した攻撃ができる。そしてショットボタンを押しっぱなしにしてエネルギーを充填してショットを放てば，より強力な波動砲を撃つことができる。もしいろんなシューティングゲームの自機同士の格闘ゲームがあったら，FORTRESS ATTACKの自機はかなり健闘すると思われる。

個人制作といってもなかなかの大作で，全部で7面＋α，キャラクターや背景のグラフィック，ステージマップもがんばっている。ボスの動きや攻撃も迫力あるし，ザコ敵の動きもトリッキーで熱い。BGMは内蔵音源のほかにSC-55にも対応しており，サウンド面へのこだわりもみられる。しかし登場キャラクターが増えると敵弾がスプライト表示オーバーで消えたりするのがちょっと怖い。また時折，処理速度がガクっと落ちることがあるのが気になった。しかしX68000で作った最初のゲームがこれだっていうんだからスゴイよね。次回はもっとすごいものを見せてくれそうだね，この作者は。

LOGICRUSH

LOGICRUSH

鴨居大吾氏作

最近では珍しい固定画面アクションゲーム。自機を操作して網を張り，それに掛かった敵を網を閉じて消滅させる，全滅させればステージクリア。どういうゲームかひと言でいってしまうと(たぶん作者はいってほしくないだろうけれど)，「平安京エイリアン」。ただ，敵を一度に複数破壊するとアイテムが出現し，自機に特殊能力が備わるという点がちょっと違う。

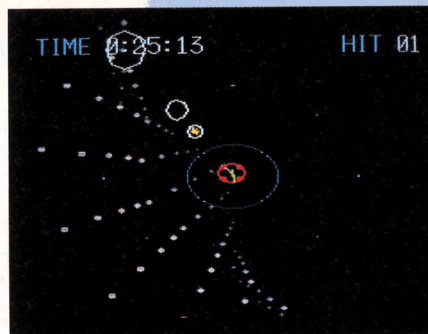
ゲームは全部で20面。ゲームスピード調整機能やマップエディタまでついている。アイテムの選択方法や敵の思考ルーチンに少々難があるものの，これだけのプログラムをX-BASICで作るとはすごい。

FormulaX

遠藤琢磨＆山本健介氏作

6人同時プレイが可能な固定画面のカーレースゲーム。全16回のレースを戦い，その総合得点で優勝を目指す。

ボタンを2つしか使わない操作系が特徴。それぞれ左右のステアリングにボタンを使用し，加速は自動的に行われる。高速時に2つのボタンを同時に押すとターボ，低速時に押すとブレーキとなる。この「同時押し」のタイミングをつかむことが勝利への糸口だ。

画面がスクロールするわけではないし，車も小さいので一見地味なゲームだが，大勢で遊ぶとなかなかやめられない。コースレコードはディスクに記録されるので，1秒以下のタイム争いも熱くなる。

コンピュータ・カーの動きもニクいので，ひとりプレイのときも知らず知らずボタンにかかる力が強くなっていることがある(必ず自分の実力とどっこいどっこいのライバルがいるのだ)。

BGMはZ-MUSIC使用でMIDI対応。対応楽器はGS音源系とMT-32系の2タイプ。

マイルームで宴会が始まってしまったときに場の盛り上げに最適かも。

GJ

麻香田美亜氏作

毎秒30コマの書き換えの超高速3Dシューティングゲーム。時間内に敵を何機撃ち落とせるかを競う。

自機をジョイスティックと加速/減速ボタンで操作，敵機を照準スコープ内に入れてロックオンしミサイルを発射。たったこれだけの内容だが，高速スクロールする地面と，自分を嘲るようにして追い抜いていく敵機の動きが，スピード感溢れる飛行感覚を演出する。これも内容が単純で結果も明確なのでカップルで楽しめそうだ。「ふぅ……ちぇっ，たった7機しか落せなかったぜ」「あきら，8機よ」「え？」「8機目は，わ，た，し」「こいつぅ……」といったスイートメモリーがこのゲームを媒介に生まれることを，この私が保証しよう。

030では動きません!?

「X68030では起動いたしません」と書いてあるが，こちらで実際に試してみたところ，Formula XはXF5キーのX68000エミュレーションモードで直接自動起動できた。その他の4つのゲームについても，必要ファイルをコピーして，Human68k ver.3.01で起動すれば動作した(もちろんデバイスドライバ類もX68030対応に変更する必要がある)。したがって「The World of X68000」についてくるゲームプログラム自体はすべて030で動作するようだ。

総合評価	0　5　10
芸術祭臨場感再現	★★★★★★★★★
お買い得度	★★★★★★★
素人パワー	★★★★★★★★★

焼きそばと麻雀ゲームの正しい関係

Kiyose Eisuke
清瀬 栄介

女３人寄ればかしましい。メンツ４人揃えば囲むのは四角いもの。そんなあなたの長い夜には，かわいい３人姉妹がお相手してくれます。勝ったらハッピー。でも負けたら，やっぱり××しなくちゃね。

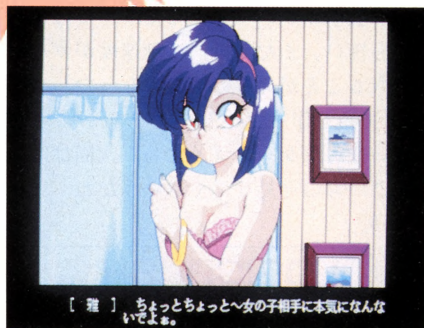
最近，ゲームセンターに女の子が増えた。それにつれて減っているのが麻雀ゲーム。エッチな絵を見つめる背中は男からみてもナニだったので，お店から姿を消すのはかまわないが，まったくなくなるのもちょっと困る。ボクは２人用麻雀のインチキくさいノリはけっこう好きなのだ。インスタント焼きそばなんかと同じで「しょせん本物にはなれないけど，本物じゃなくてこっちが食べたいときがある」という感じかな。

そこいくと，この「あにまーじゃんＶ３」などは珠玉のインスタント焼きそばの香りのする１本だ。なんといっても３人娘がガキガキ動いて脱ぐという，麻雀ゲーム界黄金のアイテム，焼きそば界の揚げ玉ボンバーを装備している。思わず期待に鼻腔をふくらますボクなのだった。

負けても負けても這い上がれ

この「あにまーじゃんＶ３」，ゲームの前に長〜いオープニングがある。内容はこのテにありがちな強引な展開だが，ノリをよくわかっていてうれしい。

３人の娘はキャラクターがはっきりと分かれている。長女の弓香は有能な秘書のイメージ，次女の雅はボーイッシュな女の子，三女の留美はブリブリだ。いちばん強いという弓香さんを選んでスタートする。

「弓香です」

性格がキツそうなとこがイイ，雅ちゃん

お，絵は悪くないし，しっかりしゃべるではないか。燃え上がるやる気。

ところが，プレイ画面を見てちょっとガックリ。画面の上がつまってて，下側が妙にさびしい。牌のデザインもややチープである。チープが身上の２人用麻雀だからって，こんなところで力を抜いちゃいかーん。

和了ったときの画面には目を奪われた。わざわざフォントを用意して，点数が伸び縮みしながら出てくるのだ。

「平和」ばーん。

「１翻　1000点」ばばばーん。

「あにまーじゃん」な迫力である。

画面に弓香さんが現れる。

「あら，お強いのね」

まずは２重スクロールでずずーっとナメていく。……待つことしばし。データの読み出しには時間がかかるので，HDへのインストールをお勧めしよう。女の子はしっかり動いて脱ぐ。最後の１枚まで見ようという気になるには十分のクオリティである。

しかし，弓香さんは強い。留美は鳴きすぎて役がなくなるほどの初心者だし，雅もなんとか競り勝ちできるが，弓香さんは強い。とにかくよくツモる。大三元字一色もツモるくせして，安めで蹴ったりもする。

「ツモ」

「きっさまー！」

負けるとスロットゲームで復活のチャンス。怒りにまかせてばかばかキーを叩き，点を稼いでふたたび卓に向かう。気持ちいー。これぞ２人用の麻雀ゲームの正しい燃え方だ！

まーじゃん界の王道を行け

このゲームのよい点はおもに３つ。

・きちんと動くアニメーション
・弓香の理不尽なほどの強さ（燃える！）
・首尾一貫した２人用麻雀のインチキくささ。これは特にプッシュしたい

逆に悪い点は，

・ツモった牌を手の中に組み込む
　お前ホントに平和なのか？
・ときどき点数計算を間違える
　６翻で満貫ってのは何だ？

この「あにまーじゃんＶ３」，あとはシステム周りの細かなところをつめれば，結構いいかもしれない。

とにかく強いのは弓香さん

X68000用　5"2HD版４枚組　8,800円（税別）
ソニア　☎03(3399)1059

このままの路線で行ってほしい

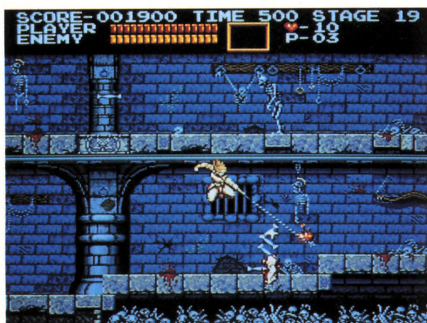
画面やツモ牌の処理のほかにも，操作性の面で頼りない点がある。捨て牌を動かすキーの動きがなめらかすぎて隣を切りやすい点などがそうだ。３人娘の個性のつけ方など，押さえるべきところは押さえてあるだけに惜しい。

X68000版には，おまけにスロットゲームがついている。こちらで高得点を達成すると，オリジナルのグラフィックが見られるぞ。

総合評価（0〜10）

操作性	★★★★★
スピード	★★★★★
特有のノリ	★★★★★★★★★

ひと言：あのときの手は絶対跳満だー！

TREND ANALYSIS

1993年9月号のハガキ集計ベスト10 最近買って気に入ったソフトは？

POINT	タイトル	発売元	発売日
243	悪魔城ドラキュラ	コナミ	'93/7/23
76	餓狼伝説	魔法株式会社	'93/7/23
29	リブルラブル	電波新聞社	'93/6/25
29	エトワールプリンセス	エグザクト	'93/3/26
29	SX-WINDOW ver.3.0	シャープ	'93/3/30
23	ロボットコンストラクションR.C.	エレクトリックシープ	'93/7/30
13	スターフォース	電波新聞社	'93/3/26
10	The World of X68000	電波新聞社	'93/8/20
10	Winning Post	光栄	'93/5/30
6	MATIER	サンワード	'92/10/9

（無作為抽出した1000通のハガキを集計）

予想にたがわず，新作ソフトへの大きな反響がすぐに出てきた。

ソフトの発売の数日後に行った先月号での集計で，すでにベスト10入りを果たしていた２つのビッグタイトル，コナミの「悪魔城ドラキュラ」と魔法株式会社の「餓狼伝説」は，ほかを大きく引き離しての１位，２位である。

そのほかにも，その直後に発売されたエレクトリックシープの「ロボットコンストラクションR.C.」が６位と健闘。さらにはそのあとに電波新聞社より出版された「The World of X68000」が８位となり，高い評価を得た新作が一挙に４本登場という結果になっている。

では順を追ってみていこう。

まず，１，２位については，注目すべきはポイント数の高さである。１位となった「悪魔城ドラキュラ」は，２位の「餓狼伝説」のなんと３倍を超えるポイント数である。しかし，その「餓狼伝説」にしても普通ならば１位になるほどのポイントを稼いでいるのだから，この２本のソフトの人気は「絶大」といっていいだろう。

３～５位の３タイトルは，先月の１～３位がそのまますべり落ちたという格好だ。「リブルラブル」は発売から日が浅いので今後のことはわからないが，「エトワールプリンセス」「SX-WINDOW ver.3.0」などは高い満足度と根強い評価を得ているようなので，今後もある程度，安定した人気を保ち続けるかもしれない。

先に述べたように，６位と８位はまだ発売間もないソフトである。「悪魔城ドラキュラ」「餓狼伝説」に話題が集まりがちな時期に登場しただけに，ちょっと不利なのではないかという危惧もあったのだが，X68000ユーザーの心をうまくとらえたようである。「ロボットコンストラクションR.C.」はパーツを選んで組み合わせることでロボットを「設計」し，命令ブロックを組み合わせてフローチャートを作成してロボットの動きをプログラミングするという「作る楽しさ」を味わえるゲームである。そして，第１回X68000芸術祭の優秀ゲーム作品を収録した「The World of X68000」。どちらも「何かを作りたい」というX68000ユーザーの志向に合った企画であるといえるだろう。

毎月集計するハガキのなかには，「ソフトを買っていない」「買ってみたが，つまらなかった」などの理由でソフト名を挙げていないものも当然かなりの枚数にのぼるのだが，この９月号のハガキには，そのようなものが少なかった。つまり，大半の人が新作ソフトを購入し，なおかつ満足することができた，ということになる。

さて，来月は，これまた人気の「クレイジークライマー/クレイジークライマー２」「コットン」のランキング入りが予想される。混戦となるか？　ポイント数はどうなるか。熾烈な競争が繰り広げられるのもまた楽しみである。

THE USER'S WORKS

●DIVE ON

スプライトビシバシ，バリバリの縦シューだ。技術的にはいうことなし。全体的な完成度は非常に高い。難易度はかなり高めかな？　TOWNSパッド，メガドラパッドにも対応している。

●

形式はオーソドックスな縦スクロールシューティングゲームである。

味方機が落としていくカプセルにより装備がパワーアップする。装備は3種類で，前方3方向から最大前方5後方2方向の弾を発射する「ワイド」，最大4ラインの強力な「レーザー」，最大2系統で敵をめがけて自機の周りをグルグル回る「サーチレーザー」の3種類がある。それぞれ，使いやすいが威力はイマイチ，強力だが攻撃範囲が狭い，便利そうだが使えない，といった特徴がある。

一般的な意見としては，レーザーが主装備でサーチレーザーはボーナス面以外使ってはいけない，となるだろう。縦シューだから当然(?)ボムもある。

オプションがあるときにはトリガーオフで自動的にため撃ちがチャージされる。ため撃ちは4段階でフルチャージ時には画面1枚分の弾が飛び出す。ボス前などところどころに合間があるので，すかさずチャージ，引きつけて一掃する。なかなかいい味を出している。

描き込みの激しいDIVE ONの画面。基本はこれらような縦長モードだが，オプションで画面一杯の横長モードも設定できる。

ちゃんとしたプログラムだから連射しても弾数が増えなかったり，弾の出が悪くなったりはしない。

もっとも弾数の多いワイドで見てみると，1画面中に最大3トリガー分の弾が表示される。1トリガーでおそらく2発の弾が出ている。さらにこれが7方向だから自機の弾だけでも画面内に42発ばらまかれることになる。

当然，敵の攻撃も激しいが，時分割表示の多用やシステムに組み込まれたスプライトダブラーによって軽々と128個を超えるスプライトを扱っている。さらに，高速なマシンでプレイすると敵キャラの爆発時に破片が飛び散るなどのフィーチャーが加わるようになっている。

完成度は十分に高いのだが，あえて欠点を挙げると，音楽がいまいちあっていない，弾避けというより弾消しが主となるのでオプションがないと非常につらい，展開は多彩なんだけどメリハリを欠く，などがある。これでキャラクターの「味」とかゲームデザインによる気持ちよさが出てくるようになるとアーケードゲーム並みのレベルといえるだろう。

入手方法

動作環境はメモリ1Mバイトで可（2Mバイト以上の場合はオンメモリで動作する)。X68030にも対応している。メディアは5インチのみなので注意すること。

このゲームを入手希望の方は，現金書留または無記名の定額小為替2000円分（送料込み）と返送用の住所を書いた宛名シール（白い紙でも可）を同封して下記住所まで送ってほしい。

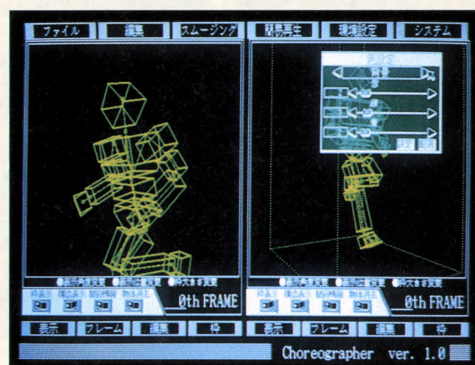
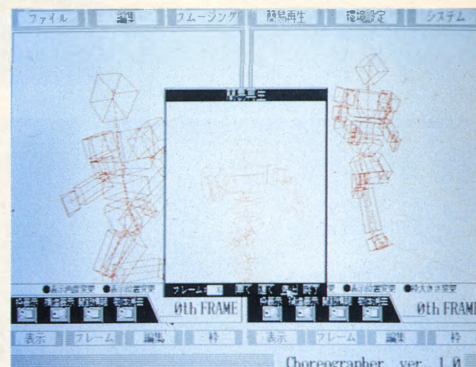
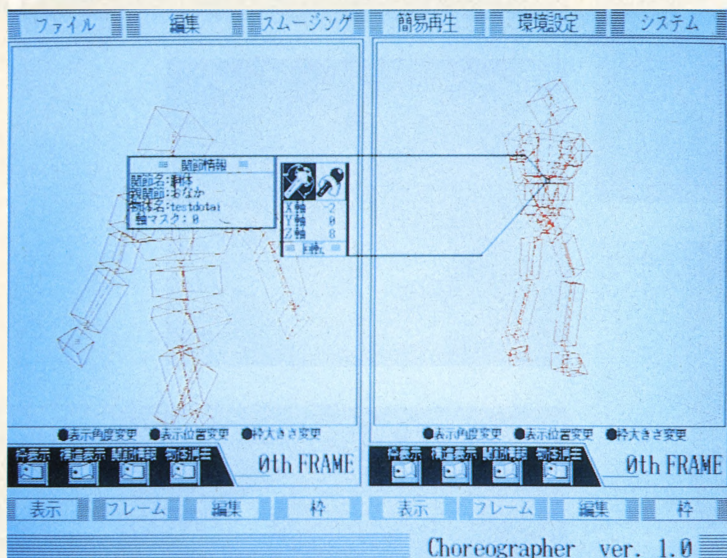
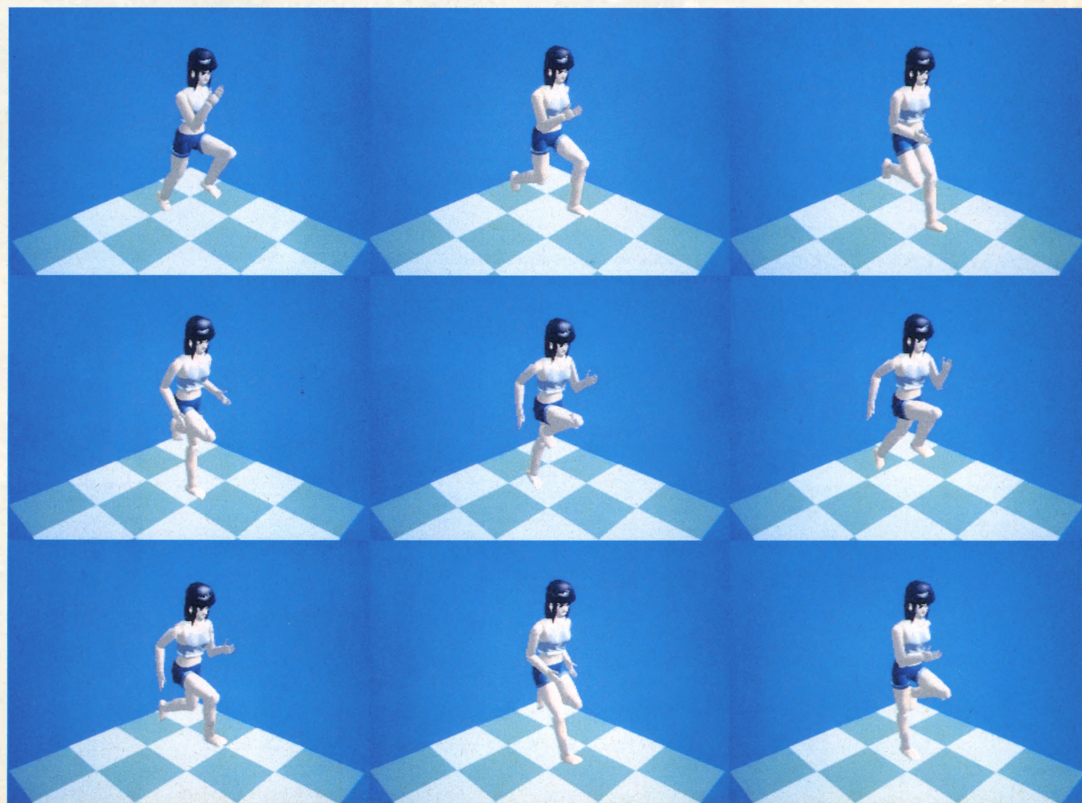
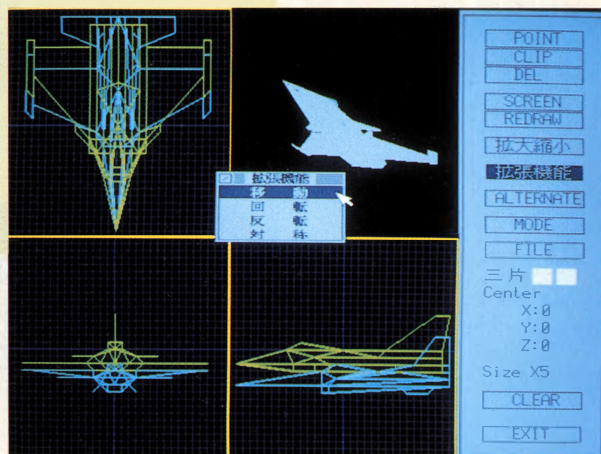
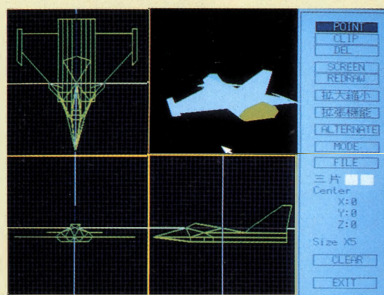
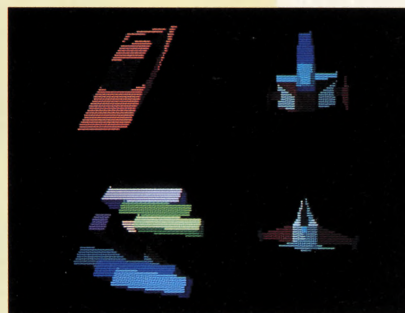
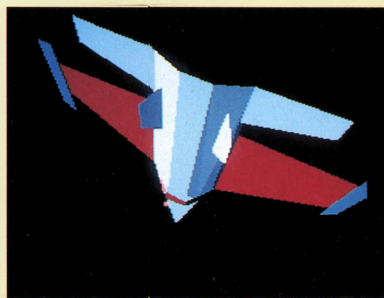
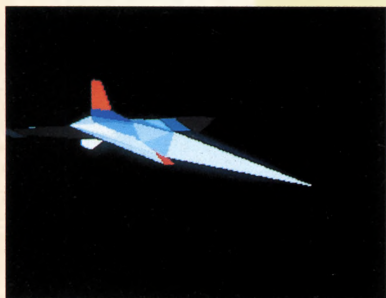
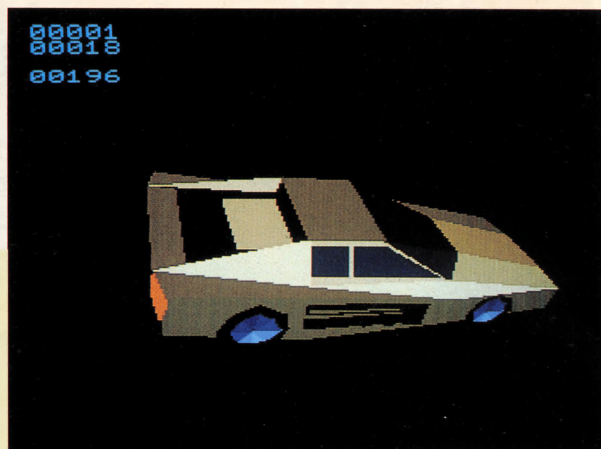
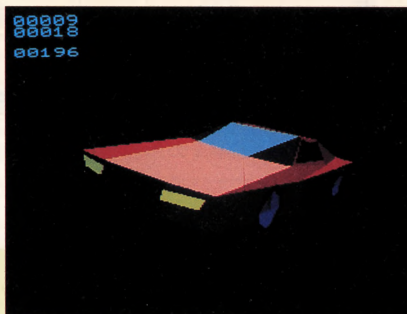
〒202　東京都保谷市富士町3-2-3
青柳荘1F5号室渡辺方
MIXダイブオン通販係

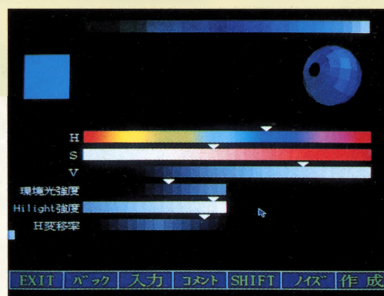
キャラのアニメーションが少ないのがコンシューマっぽいのだが，十分に派手な画面を持つ。細かい部分にはかなりこだわって作ってあることがわかる。

【特別企画】

どどーんと秋祭りPRO-68K

1年と3カ月ぶり，Oh!Xの付録としては1年4カ月ぶりの付録ディスク
今回はややゲーム指向で大物指向
収録内容をどどーんとカラー写真でお見せしましょう

Choreographer

Choreographer（CGR.X）はDōGA CGAシステムで使用できる汎用の構造体モーションエディタだ。さまざまな形状データを指定して読み込み，任意の関節を曲げてポーズをつけ，用意された2種類のフレーム間を滑らかに補間していくことができる。
右の写真はサンプルで付属するデータをTAKERUで販売されている人体モデル集のデータに適用してレンダリングを行ってみたものだ。
重心の移動などがまだサポートされていないので，多少不自然な部分もあるが，これくらいのデータ作成ならきわめて簡単な操作で実現できるようになっている。ぜひ活用してみてほしい。

SLASH

今回の付録ディスク中いちばんの大物。X68000用ポリゴナイザライブラリSLASHのサンプルプログラム群だ。なんとコア部分のソースだけでディスク丸1枚，開発期間3カ月で完成したソースは17000行を軽く超える。うーむ，横内威至おそるべし。
ちなみにサンプルのテスタロッサは開発時の標準モデルとして使われたもの。
右はSLASH開発サポートツールのアトリビュートエディタと簡易モデラだ。上は450ポリゴンの物体を乱数で地形化したもの。

PENJANG!

1993年6月号で発表された簡易麻雀ゲームの改良版。役が増え，思考ルーチンが多彩になった。ズルはしていないが，なめてかかると全然勝てなかったりするので注意。

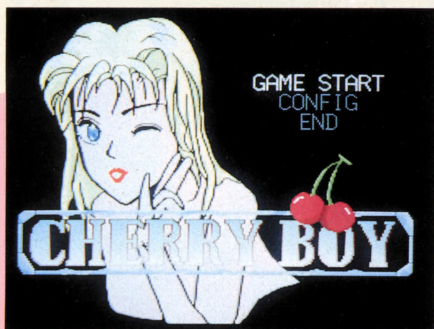
Dracula Clock

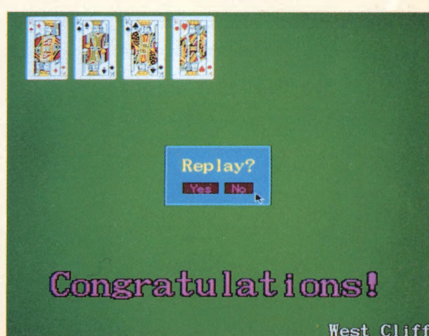
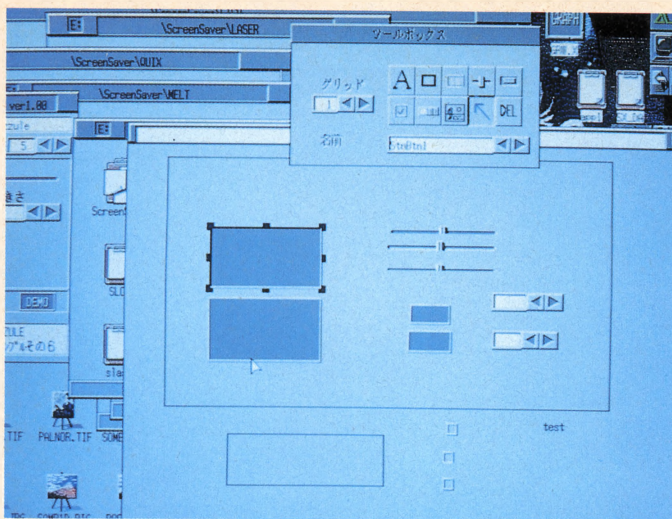
あの時計台が再現される。秒針つき。

SION IV DEMO

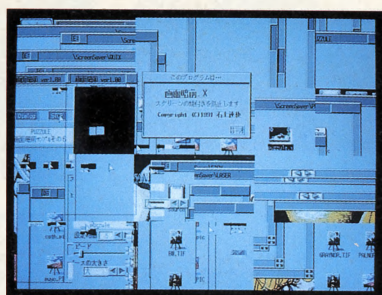
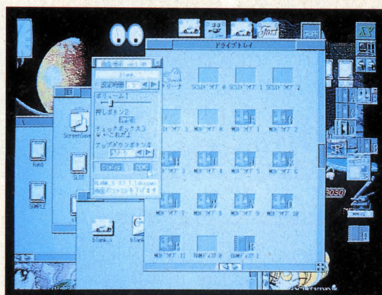
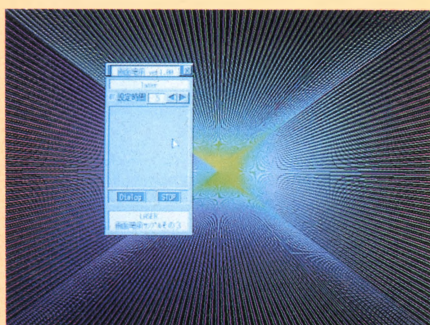
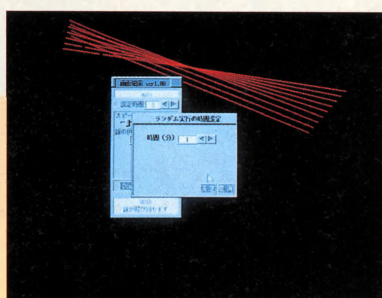
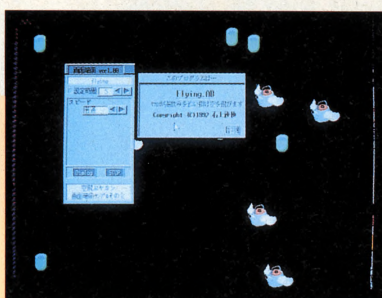
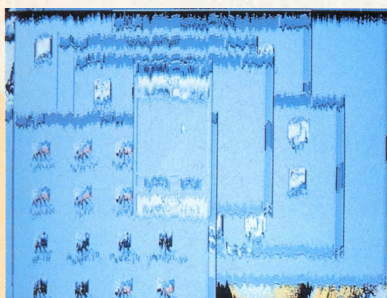
SIONシリーズ最新作はSLASHを使用したポリゴン仕様の3Dスペースシューティングだ。まだまだサンプル版とはいえ，怒濤の敵編隊，巨大戦艦，砕け散るアステロイドといったフィーチャーを盛り込んでいる。今回は時間管理により，ゲーム展開の速度は固定されている。高速なマシンほど滑らかな動きになるぞ。

CHERRY BOY

なんかあぶない名前のパチスロシミュレータ。ドラムが大きくなって操作性は向上している。連チャンで日頃のうっぷんばらしだ。

WEST CLIF

CARDDRV用のサンプルプログラムはクロンダイク風のWEST CLIFだ。右のように場に並べられたカードを赤黒交互になるように順番に移動していって，それが下のように積み重ねられ，最終的に右下のようにまとまったらあがりとなる。実際にやってみたほうがわかりやすいか？

Window Designer

SX-WINDOW用の簡易ウィンドウエディタもどき。完成版ではウィンドウがC言語のソースプログラムとして出力される予定だが，そのあたりは開発キット待ちといったところか。

Screen Saver

SX-WINDOW用のスクリーンセーバーシステムとそのモジュール群だ。いまや世の中のスクリーンセーバーは純観賞用といってもよいくらい派手なものになっている。そこで，FISHである。このモジュールはSX-WINDOW環境では埋もれていたX68000のハードウェア資源を目一杯使っている。

【特別企画】

秋祭りPRO-68K

実に久々の付録ディスク。
X68000の世界では流通するフリーソフトウェアも粒揃いで，ユーザーの環境はずいぶんよくなった。ソフトウェアの完成度も高い。パソコン通信をしていなくてもそういったものが簡単に入手できるようにもなってきた。
そういった状況のなかで「Oh!Xの付録ディスク」というのはどのような位置づけをされるのだろうか。
単なる1枚のフロッピーディスクも，使い方次第で素晴らしいものにも，つまらないものにもなりうる。しかし，本当に望ましい方向性を示すことは実に難しいことである。
今回の付録ディスクはなんやかんやで前人未到のフロッピーディスク6枚組に展開される。ちょっとでっかいモノがあったため，いろいろなところにシワ寄せがいってしまったが，まあ，それはそれ。
X68000の文化は成熟の時期にある。ここでさまざまな人の努力による成果を反映することも重要だが，新しい　　を開拓することはもっと重要であろう。すべからく「祭り」とは，成果を奉じ，未来を祈念するためのものなのだから。

［CONTENTS］

付録ディスクの内容について

編集部

さっそく付録ディスクに収録されたプログラム内容を見てみましょう
とりあえず，フロッピーディスクに展開します
6枚組になりますので必要なものから選んで始めるのがいいでしょう

お待たせしました。1年と3カ月ぶりのOh!X付録ディスクとなりました。本当なら春には出したかった付録ディスクなのですが，諸々の都合で秋にまでずれ込んでしまいました。

最近はどうもアプリケーションやシステムは巨大化しがちなものが多く，今回の付録ディスクでは，削れるものはすべて削ってディスクに目一杯押し込んでも以前ほどのバラエティというわけにはいかなくなってきました。その分大型のプログラムには気合が入っていますので，皆さん存分に活用してください。

今回は一応，要2Mバイト以上ということになっていますが，大半のものはメモリ1Mでも使用できます。また一部のものは事実上メモリ4Mバイト以上あることが望ましい場合もあります。メモリが足りない状況については各自で対応してください。

それではさっそく今回のディスクに収録されている内容について解説していきましょう。

各ディスクの内容

付録ディスクの中身は圧縮されており，指定された手順で操作することにより，フロッピーディスク6枚に展開されます。各ディスクの内容は，

●ディスク1
内容　各種ツール

DōGA CGAシステム用構造体エディタChoreographer，SX-WINDOW用ウィンドウデザイナなどをはじめ，その名のとおり各種ツール群を集めたディスクです。詳しくは囲みを参照してください。

●ディスク2
内容　SX-WINDOW用スクリーンセーバー

これはSX-WINDOWアクセサリ集にあったシャープ純正のスクリーンセーバーではなく，以前に紹介した「石上版」というやつです。サンプルモジュールとして1993年5月号で作成したものと同じものと，メモリ2Mバイトではちょっときついかもしれないという弩級モジュールFISH.Xが用意されています（動かないことはないが）。

もちろん実行にはSX-WINDOWが必要です。当然ある程度メモリもなければ実用にはなりません。ご了承ください。

●ディスク3
内容　SLASHソースリスト

ポリゴナイザライブラリSLASHのコア部分のソースリストです。ディスク容量の都合上，ライブラリそのものは収録できませんでした。ライブラリのオブジェクトはmakeを使うか，makefileを参考に各自で作成してください。

具体的にはディスク6の内容と合わせ，一般的な開発環境の整ったハードディスク上に展開します。するとそのままで推奨環境ができあがります。あとは，

　make depend

と入力するだけです。プログラムの性格上，ある程度の開発環境がなければ使用できないものです。あしからずご了承ください。

●ディスク4
内容　各種ゲーム

簡易麻雀PENJANG!，スロットマシンCHERRY BOY，さらにCARDDRVとCARD2.FNCのシステムおよびサンプルゲームWEST CLIF（作：高山忠信，ルールはクロンダイクに似ています），悪魔城ドラキュラの時計台をそのまま時計にしてしまうツール（悪魔城ドラキュラが必要です）などが収録されています。

●ディスク5
内容　SION IVデモバージョン

SLASHシステムを使用した3Dシューティングゲームのデモです。ただし，無敵バージョンですのでまだゲームにはなっていません。バランスやキャラクターなども暫定的なものと思ってください。なお，展開前に創刊10周年記念PRO-68Kに収録されていたSION IIのディスクを用意しておくとSION IIのBGM，効果音を流用します。

なお，このディスク5以外のディスクにはシステムが転送されませんので注意してください。

●ディスク6
内容　SLASH用ツール＆サンプルプログラム

SLASH用のツールと使用サンプルなどです。基本的にディスク3と対になるものです。サンプルプログラム（ALLTESTA.X)を見る以外のことをするには，ディスク3に収録されたソースリストからSLASH LIBを作成する必要があります。その際には，

　アセンブラ（AS.XまたはHAS.X）
　リンカ　　（LK.XまたはHLK.X）
　アーカイバ（AR.XまたはHAR.X）

などが必要になります。

開発環境としてハードディスクと4Mバイト以上のメモリがほぼ必須です。RAM2Mバイト時には必要のないドライバをすべてはずしフリーエリアを目一杯広げたシステムディスクが必要になります。うまくいかない場合には素直にRAMを増設してください。

付録ディスクの展開について

今回のディスクは1.4Mバイトのフォーマットで作成されています。ですから，通常のシステムから今回の付録ディスクをアクセスしようとすると，

　「無効なメディアを使用しました」

というエラーが発生するはずです。必ず今回の付録ディスクからそのまま起動するようにしてください。

今回はVS2やDSHELLなどのビジュアルな操作環境を取り入れるだけ余裕がありませんでしたのですべてバッチファイルで

処理されます。

●Oh!X推奨の方法

通常のHuman68kでフォーマットしたディスクを6枚用意したうえで付録ディスクを立ち上げます。あとは画面に表示されるメニューに従って操作し6枚のディスクを展開してください。

とりあえずハードディスクなどへ直接展開したいという方でも最低1枚ブランクディスクを用意することをおすすめします。起動後にバッチファイルを終了し，RAMディスク(G-RAMにとってあります)経由で9SCDRV.Xを転送すればディスクなしでも展開はできますが，そのあたりは各自で行ってください。

システムについての注意

今回の付録ディスクのシステムには，最新のHuman68k ver.3.1+COMMAND.X ver.3.0が搭載されています。よって，もちろんX68030でも従来機種でもそのまま起動することができます。

Human68k ver.3.0以降をお持ちでない方でシステムのバージョンアップをしたいという場合には，別途Human68kver.3.0以降対応のシステム周りのデバイスドライバなどが必要になりますので，このディスクだけでは対処できません。ハードディスクのシステムを書き換えてしまった場合などでは最悪の場合，ハードディスクからの起動ができなくなることも考えられます。注意してください。

各種ツール群について

それでは今回のディスクに収録された各種ツール群を紹介しよう。どうしても必要になるもの，あると便利なものをみつくろってみた。

フリーソフトウェアで出回っているツールとしてLHA.X, BDIF.X,BUP.X,9SCDRV.X, 編集室内で使われているツール，付録ディスクの起動に必要なツールなどがまとめられている。

付録ディスクでフリーソフトウェアの配布をもっと行ってほしいという声も多いのだが，磁性面の制限からなかなか手が回らないのが実情である。

いまさらいうまでもなく，X68000のフリーソフトウェアには多種多様なものがある。ほとんど不可欠になっているものから，あると便利なもの，人によっては非常に便利に使えるもの，など実にたくさんある。そういったものについてはパソコン通信で入手するのが正道であろう。サポートなども迅速に行われる。

最近では通信をやってない人でもフリーソフトウェアを集めた書籍やディスクマガジンを購入するなり，あるいはTAKERUなどを通しても入手できるようになっているので各自で有効に活用してみてほしい。

各ツールについて

●LHA.X

毎度お馴染み，ファイルを圧縮するために使用している書庫管理ツール。相変わらず強力な圧縮率を見せてくれる。今回はディスク6枚への展開が行われる。使用法はヘルプメッセージあるいはドキュメント参照のこと。

●BDIF.X/BUP.X

X68000でほぼ標準的に使用されているバイナリ差分管理プログラム。デバッグやパッチ当てなどで今後使用されることになる。使用法など詳しくはディスク内のドキュメント参照。

●MAC.X

Oh!X標準のマシン語入力ツールだ。8ビット時代から使われていたMACINTO-Cという，いまとなっては恥ずかしい名前のツールをもとにしたものである。表示内容や操作体系もMACINTO-Cをほぼ継承している。

X68000版はリスト入力の都合からX-BASICで簡潔に記述されていたので，ちゃんとしたメモリの確保などというのはできなかった。

ということで，今回のバージョンでは，とってもみっともなかったメモリ制限の64Kバリアを突破した。といっても無制限なわけではなく，上限が8倍になっただけ。それも静的な配列で確保されているので扱いやすいものではない。たった512Kバイトだが，世の中には600KバイトくらいのフリーエリアしかないDOSだってごろごろしているのだから贅沢はいわないことにしよう（そのうち改善するか？)。

また，X-BASICで記述されたプログラムであるにもかかわらずコンパイルを前提にしているのでX-BASIC上では動作しない。

追加機能や変更された仕様は以下のとおり。

・エディットモードが基本モードになった。
・漢字表示がちゃんと行われるようになった。
・ファイル名がコマンドラインから渡されるようになった。
・扱えるデータサイズが512Kバイトまでに拡大された。
・ESC+アドレスで指定した場所を含むブロックが表示されるようになった。
・セーブ時にサイズ指定以外に，現在のカーソル位置以前あるいは以降をセーブするモードが追加された。

●9SCDRV

今回のディスクを1.4Mバイトフォーマットにするために使用したドライバ。今回のディスクでは，

80トラック
1024バイト/セクタ
9セクタ/トラック

の2HS形式が採用されている。このフォーマットで作成されたディスクは9SCDRVを組み込んだシステム以外ではアクセスできないので注意すること。

ドライバの組み込みには，

9SCDRV

と直接実行する。これでこのドライバがサポートしている2HS，2HDEなどのディスクフォーマットが扱えるようになる。

2HSなどでフォーマットされたディスクを作成するには9SCFMTを使用する。詳しい使い方はディスク内のドキュメント参照のこと。また，X68030には完全対応されていないようなので，使用中に問題があったら使用を中止することが望ましい。

●DCON

ディスクをイジェクトするツール。

DCON A:

のように使用する。詳しくはヘルプメッセージ参照のこと。

●DRV

最初に見つけたRAMディスクの位置を環境変数“ramdisk”に格納する。そして格納された環境変数を，

PATH %ramdisk%

のように使用する。

詳しい用法は付録ディスクのAUTOEXEC.BAT参照のこと。

●WILD

ワイルドカードを展開するためのツール。ワイルドカードが使えないコマンドの直前につけて使用する。たとえば，

WILD ZMUSIC -C ＊.ZMS

のようにする。

今回はZMUSIC.Xでのコンパイルに使用している。まあ，なくてもなんとかなるが，あればなにかと便利であろう。FORコマンドよりは使いやすい。

●DISH

任意のバイナリファイルをアセンブラソース形式に変換するツール。横内君御用達。

●MODRV.SYS

IBMフォーマットの3.5インチMOを読むためのドライバ。たいていの場合は，CONFIG.SYS中に，

MODRV.SYS /ID2,0

のように指定することで使用できる。IBMフォーマットのMOがないときにはなんの役にも立たないので注意。詳しくは1993年9月号を参照のこと。

●SAVESC.SYS

1992年11月号で掲載された画面保存プログラム。SRAMに登録して使用する。具体的な使い方は62ページ参照。詳しくは1992年11月号参照のこと。

●CV_SAVESC.X

SAVESC.SYSで保存されたグラフィック画面の情報をAPIC形式で取り出すツール。制作は佐藤正治氏。

SAVESC.SYSで作成したSCREEN.DATを用意し，コマンドラインからCV_SAVESC.Xを起動する。画面にメッセージが出るので，G,H,J,K,T,Pのいずれかのキーを押す。G～Kはグラフィック画面の0～3に相当する。Tはテキスト画面，Pはグラフィック画面1024×1024ドットモード時に画面の縦横比を変更してセーブするための指定である。ついでマウスをドラッグして矩形範囲を指定し，画面の指示に従ってファイルネームを入力する。カレントディレクトリにあるSCREEN.DATが参照されるので注意すること。

DōGAの振付師
Choreographer

Shibata Atsushi 柴田　淳

**できることはかなり凄いとわかってはいても
やはりちょっと取っつきにくいのがDōGA CGAシステム
CGR.Xは操作環境を変えてくれるかな？**

DōGAのCGAシステムを手にし，ひと通り使ってみてこう思った。「ここからよくあんな作品たちができあがるなあ」

たとえば，美しいカーブを車体に抱くTORNADO。DōGAのシステムを使ってみる前は，あれはきっとトテツもなく高性能のモデラかなにかで作られているのだと思っていた。

ところがどうやら，現実は違うらしい。以前本誌に載った文月さんの記事を見ると，あの車体は紙上の設計図から起こされたものということだ。紙の上から座標を読み取り，座標を1つひとつ手作業で打ち込んでいく。

僕は元来面倒くさがりなので，世の中にそんな七面倒くさいことをやる人がいるというのが信じられない。それだけ文月さんの映像制作にかける情熱が凄まじいということなのだろうが，僕の場合は思考回路がスレているので，情熱が行動に転化しない。

じゃあそんな人間は，CGAを作れないのか。アイデアはあるが，幾多の障害を乗り越えてそれを実現させる情熱がともなわない者は，映像制作からは爪弾きにされる運命にあるのだろうか。

詰まるところ，DōGAのシステムは敷居が高すぎるのである。システム中のアプリケーションは，それぞれ粒揃いでよくできてはいる。しかしその半面，相互の連係が取れていないような気がする。アプリケーション群を使いこなして作品を仕上げるには，それぞれのプログラムの癖を見抜き，その癖を踏まえたうえで取りかからなければならない。

こんなのも簡単

CGR.Xの基本画面

「いまどき統合的なGUIを持たないCGAシステムなんて時代遅れさ」

なんていうわけ知り顔の意見を引き合いに出す以前に「全体として取っつきにくい」というDōGAの性格は，敷居がもっと低ければ参加できるであろう素晴らしい才能たちを遠ざける原因になってはいないだろうか。

僕がX68000を買った目的のうち3分の1は，実はCGAの制作にあった。しかし，前述したように現在のDōGAのシステムを使ってCGAを作る気力は，残念ながら僕にはない。

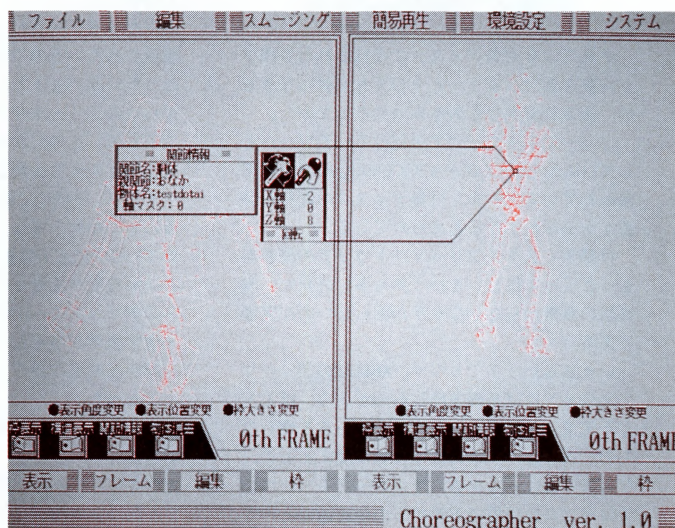
ではAMIGAでも買おうかとも思うが，それではなんだか負け犬みたいでいやだ。だいいちお金もないし。

そんじゃあどうするか。いろいろ考えているうちに，天から声が降りてきた。

「ないモノは作れ」

CGR.Xの思想

ところで，GUIの素晴らしさはどこにあるのか。たとえば，ディレクトリ上にあるファイルの属性を，つまり問題にするファイルがテキストであるのか，実行形式であるのかなどといったことを判別する場合を考えよう。

DOS上からファイル属性を確認するためには，拡張子を見たり，ATTRIBでファイル属性のリストをとったりする。あるいはビュアーで直接内容を覗いて見たり、ヘッダで判別するなんていう方法が考えられる。

ただ，ここで問題が生じる。というのは，DOS上でファイル属性を確認する方法は，すべて「知っていなければ判別できない」方法ばかりなのである。つまり，DOS上でしか有効でない浮き世ばなれした法則を，わざわざ覚えなければならない。

ところが，テキストファイルには紙の，実行形式のファイルには汗をかいたコンピュータの絵を割り振り，画面に表示するとする。ユーザーはその絵を見て，ファイルの属性を類推することができる（適切なアイコンが定義されていると仮定しての話ではあるが）。

つまりGUIの利点とは，

「ユーザーがすでに知っている事柄とコンピュータ上の概念を結び付ける」

ことにあるのだ。ユーザーは，少々の決まりごとを除いてほとんどなにも新たに覚える必要はない。

現在のDōGAの状況は，いってみればいまだにDOSシステム然としているのだ。新たに覚えなければならないことが多すぎ，それが取っつきにくさを誘発しているように思える。

去年のBASICの特集でも似たようなことを書いたけど，CGAを作る人にとっては，物体の座標とか，フレーム中の座標位置などといった「数値的概念」は直接には関係のないものである。むしろその数値によってどの位置に，どのくらいの大きさで物体が描かれるか，というようなことのほうを気にすべきなのだ。

しかし現在のDōGAのシステムでは，まず数値を入力しそこからどのような画像が得られるかを想像しろと強いる部分が多すぎる。このようなことを人間がするのは不可能だとは思わないが，相当な困難をともなうことは確かだろう。特にモデラ，フレームエディタなどは早急により高機能なものへと移行しないと，DōGAの存続にも関わるのではないか。

そこで「ないモノは作れ」なのである。とはいっても，まだホンのとっかかりにすぎないが。

Choreographerって?

コレオグラファと読む。元来の意味はバレエの振付師のことらしいが，今ではバレエでなくとも振付師は押し並べてこう呼ばれる。たいていの英和辞書には，動詞の「choreograph」として載っているだろう。ただし，

「ラッキィ池田は離婚ダンスをコレオグラフした」

という日本語の用法があるかどうかは定かではない。

さて，CGR.Xは，DōGAの構造体のポーズおよび動きを編集するためのソフトである。てっとり早くいってしまえば，人間とか犬とか象とかの複数の関節からなるもののアニメーション作成を支援するもの。

操作方法の詳細は別記のマニュアルに譲るとして，ここでは操作の流れをかい摘んで見ていくことにしよう。

まず，付録ディスクに添付のMAN.STRを構造ファイルとして指定し，CGR.Xを立ち上げてみよう。しばらくすると，編集画面が現れる。

人間の描かれた矩形の右下を見ていただきたい。フレーム番号が0番となっているはずだ。そのことをとりあえず確認しておき，今度は人間のポーズを変えてみよう。

ウンコ座りから

関節を回転させればいいのである。

手始めに，俗にいう「ウンコ座り」をさせてみよう。大腿部，膝，などの関節を動かせば，それほど苦もなくウンコ座りが完成するはずである。

次に編集するフレームを変更する。先ほどのウンコ座りが0フレームだったのを思い出し，こんどは20フレーム目を編集しようか。第20フレームに移行すると，初期のポーズをとった人間が現れるはずである。そこは両腕を上げて背伸びをしている格好にしてみようか。

2つのポーズが完成したら，画面上部のメニューでスムージングを選ぶ。0フレームから20フレームまでをスムージングすると，その間のフレームが中割りをしたポーズで埋められる。

スムージングをしたら，簡易再生で動作を確認する。すると，立ち上がるときの手の動きが不自然だということに気づく。

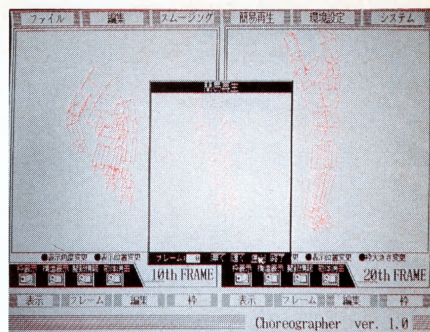
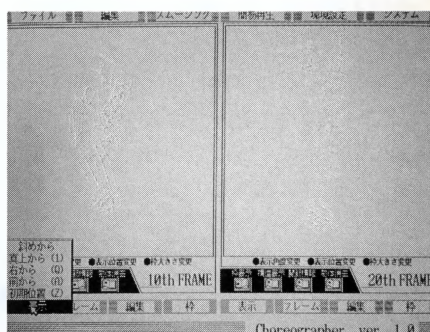
再度フレーム変更。10フレーム目を呼び出し，肩，肘の関節を調整する。あとはスムージング，動作確認をし，納得のいくまでこれを繰り返す。

これがどんなに複雑な動作であれ，CGR.Xでの編集作業はだいたいこのような感じで進んでいくと思っていい。いま例に挙げたような作業なら，5分とかからず終わるはずである。ウソだと思うなら計ってみるといい。

今後のこと

CGR.Xは，なによりも「使って楽しい」モーションエディタを目指して作った。誰にでも，ストレスなく構造体のポーズの編集ができるように，というお題目はとりあえず満たしたと思う。

プログラムはCで書かれているが，肝心なところ（座標の回転など）はアセンブラで記述し，最速とはいかないまでもそこそこの速度を実現している。

また，メモリの節約も考慮しており，メ

背伸びをする

おかしいところを直す

モリ1Mバイトのマシンでも70フレームほどの簡易再生ができる。これは，簡易再生のための画像バッファにG-RAMを使っているためなのだが，逆にいうと描画はすべてテキスト面で行っている，ということになる。

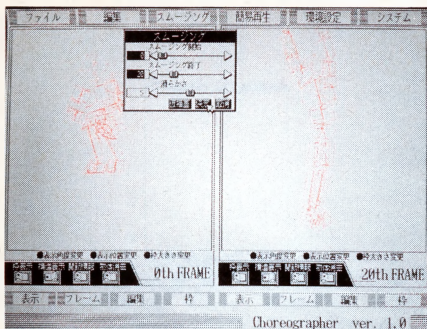
とはいえ問題点がないか，というとそうでもない。まずいちばんの問題点はCGR.Xの書き出すファイルのこと。

CGR.Xはファイルを保存する際，DōGAのフレームソースとして書き出す。その中に配列を設け，配列の初期値の形で回転角を書き出す。

で，これをDōGAのフレームファイルに落とすにはどうするか。熟練者なら，

「配列に回転角が書き込まれているなら，その配列から角度を得る関数を書けばいいんじゃん」

と考えるだろうが，初心者はそうはいかない。要するに初心者にとってCGR.Xは，「ワイヤーフレームの人間を動かして遊ぶ」くらいの役にしかたたないのだ。CGR.Xによって，DōGA CGAシステムの敷居は低くはならないのである。

DōGAに依然求められているのは，すべての作業を，いやせめてフレームエディットをオンシステムでこなせる統合環境なのである。いま僕の中では，CGR.Xはその小手調べにすぎず，いずれ大幅なバージョンアップなり新しいソフトを作るなりしなければならないのだ，という思いが湧き上がってきている。

スムージングの指定

DōGAのスタッフの皆さんは，おそらく学生であったり，社会人であったりと，本業を持ちながら片手間にシステム開発をなさっているのだと思う。すると開発に割く時間も限られてくるだろうし，大掛かりなシステム開発は困難なのだろう（分業であればなんとかなるかもしれないが）。

で，僕のように，暇で，そこそこプログラムも組めるような人間がいるとして，さらにそいつがなんらかの問題意識を持っているとして，なにもしないのはイカンことだ。そんな思いから作り始めたのがこのCGR.Xだった。

おそらくたいていの人は，CGA制作という手段を通してDōGAと関わりを持つのだと思うが，その一方で，アプリケーションの開発という手段を通して関わりを持つ者がいたっていいじゃないか。映像制作者がDōGAで凄い作品を作るのの代わりに，いままでにないアプリケーションを引っさげてスタッフを刺激するのである。これが僕なりのDōGAへの関わり方であり，協力のしかたなのである。そうだそうだ，そう決めた。

最後に

マニュアルにも書いたが，CGR.Xはフリーソフトである。本体，ソースとも商行為に用いない限り，使用制限を設けない。だいいち，このソースを解析して理解できるほどの実力のある人は，これと同じかそれ以上のものを作れるはずである。

なお，CGR.Xは

GCC.X　真理子バージョン　X6_SX01
HAS.X　version 2.35
HLK.X　version 2.22

を使用して開発されたものである。それぞれのツールの作者の方々には，この場を借りて深くお礼申し上げたい。

DōGA構造体モーションエディタ Choreographer ver 1.0　マニュアル

1 Choreographerについて

・Choreographer（以下CGR.X）は，X680x0上でDōGAの構造体のポーズを編集するためのソフトである。

基本的に，DōGAシステムのアプリケーションMFE.Xと同等か，それ以上のことを実現できる。以下に特徴を箇条書きにすると，

・操作はフルマウスオペレーション
・インタラクティブな構造体編集
・高速なワイヤーフレーム画像出力
・オンシステムで動作確認が可能
・DōGAの構造体を，ほぼそのまま編集用の構造ファイルとして取り込める

という具合になろうか。特に，操作性には相当気を遣って作ったつもりだ。速度面も含めて，10MHzユーザーでもストレスなく構造体のポーズの編集ができるはずである。

2 起動方法

起動には，実行ファイルであるCGR.Xのほかに，任意の構造ファイルとDōGAの形状ファイルが必要である。

編集したい構造体の構造ファイルをカレントディレクトリ上に用意し，コマンドラインから，

CGR /SMAN.STR

などとすることでCGR.Xが起動する。

起動オプションは，スラッシュ（/）またはハイフン（－）のいずれのあとに記述しても認識される。

使用できるオプションを並べると，

・Sオプション：構造ファイル名を指定する。拡張子を省略した場合は.STRとなる。

.FRMなどと，異なった拡張子を特に指定することで，DōGAのフレームファイルなどを構造ファイルとして読み込める。

・Fオプション：編集する最大フレーム数を指定する。デフォルトは100。特に大きな値が必要な場合，またはメモリの関係でデフォルト以下に縮めたいときに使うとよい。

・？オプション：起動方法の説明を表示する。

3 操作方法－1

a）CGR.Xの画面構成

CGR.Xの起動時の画面は，大きく4つの部位からなる。

・画面上部に並んだメインメニュー
・画面中ほど左の1つめの構造体編集画面
・同じく右の2つめの構造体編集画面
・左右の編集画面下にあるサブメニュー

特徴として，構造体編集用の画面が2つ用意されている。左右別々のフレームを編集することもできるし，同一フレームを違った角度から同時に見るということもできる。後者の場合，片方の構造体の編集結果は当然もう片方の画面にも反映される。

b）構造体のポーズの編集

CGR.Xでは，構造体の編集にインタラクティブな操作系を採用している。

・CGR.Xで構造体のポーズを編集するためには，画面に表示されている構造体の動かしたい関節の上にマウスカーソルを持っていき，マウスの左ボタンを押す。

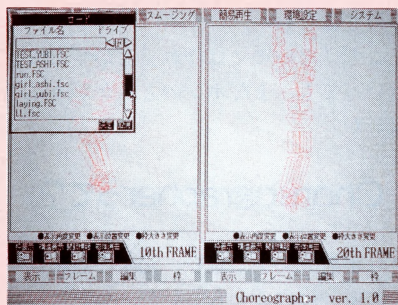
・ボタンを押したままの状態でマウスを移動させれば，目的の関節のX・Y軸の角度が変わる。

・また，左ボタンを押したままの状態で右ボタンも同時に押せば，Z軸の角度が変わる。

・ただし，関節の角度を変えるときには必ず「左ボタンを最初に」押さなければならない。

なお，この操作方法は，関節に限らず，CGR.Xにおいて角度を変更する場合にはすべて共通している。

構造体のポーズを変更すると，編集画面右下のフレーム番号に「下線」がつく。これは，ポーズの中割りの際のキーフレームとなったことを表している。

以後このようなフレームのことを「標本値」と呼ぶことにする。

c）構造体の表示角度の変更

編集中の構造体を，いつもは正面から見ているが，ときどき確認のため違った角度から眺めたいと思うことがあるに違いない。

・構造体の表示角度を変えるには，構造体が表示されている領域の下の部分にある，「表示角度変更」と書かれた左の黒丸の上で左ボタンを押す。

・角度の変更方法は，3-bの構造体の編集方法に準ずる。

d）構造体の表示位置の変更

表示角度と同様に，細かい部分をズームアップするなど，構造体の表示位置を変えたい場合もあるだろう。

・構造体の表示位置を変えるには，編集画面の下部にある「表示位置変更」と書かれた左の黒丸の上で，マウスの左ボタンを押す。

・左ボタンを押した状態でマウスを前後に動かせば，表示位置が前後に移動する。

・左ボタンとともに右ボタンも押せば，上下左右に移動する。

・角度の変更と同様に「右ボタンを必ず最初に」押さなければならない。

この操作方法は，座標値などを変更する操作に共通したものである

e) スイッチの操作

構造体編集の補助的かつ簡易な機能を，表示画面下に並んだスイッチに割り当ててある。

たとえば，構造体のだいたいの大きさを示す「枠」の表示，非表示であるとか，構造体の骨組みを表示する，などといった機能を，スイッチのオン，オフで切り替えることができる。

・スイッチのオン，オフは，その上で左ボタンを押すことにより交互に切り替わる。

また，スイッチには4種類がある。

・枠表示：構造体の大まかな外核を示す「枠」の表示，非表示を決定する。

・構造表示：構造体の骨組みの表示/非表示を決定する。関節の位置が塗り潰された小さな矩形で表現されるので，構造体編集の際，関節の位置の目安となる。

・関節情報：マウスカーソルで示された，関節の情報の表示・非表示を決定する。

・物体消去：構造体の表示を骨組みだけとし，表示の高速化を図る。

f) 枠の大きさ変更

・枠の大きさを変えるためには，表示画面下の「枠大きさ変更」の左の黒丸の上でマウスの左ボタンを押す。

・変更中のマウスの操作については3-dの構造体の表示位置変更の方法に準ずる。

4 操作方法ー2

a) ポップアップメニューの操作

編集に必要な機能のうち大部分は，ポップアップメニューに割り当ててある。

・ポップアップメニューを開くには，目的のメニュータイトル部の上でマウスの右ボタンを押す。

数値選択用としてスライドボリュームが用意されている。このスライドは以下のように操作する。

・スライドの左右のボタンの上でマウスの左ボタンを押すことにより，数値を細かく変化させることができる。

・スライドの上で右ボタンを押せば，スライド自体を動かすこともできる。

・フレーム選択の際，選択数値の表示が反転している場合，そのフレームは標本値であることを表している。

標本値については2-bを見返していただきたい。

スライドボリュームのほかに，任意の操作をうながすボタンがある。メニューの操作で指定した事項を決定するか，または取り消すかを選択するためのものなどである。

また，ボリュームによってフレーム数を決定する必要のある操作の場合，目的のフレームの構造体のポーズを見たい場合があるかもしれない。そのような場合には，仮描画というボタンも用意されている。

・決定ボタンの上でマウスの左ボタンを押すと，選択した処理が実行される。また取り消しボタンを押すと，選択した処理を実行しない。

・仮描画のボタンを押すと，編集画面に選択したフレームのポーズが表示される。ただし選択したフレームが2つの場合，左右の編集画面に分けて表示する。

b) 編集画面のメニュー

構造体の関節の近くでポップアップメニューを開くと，関節の編集のメニューが現れる。

・退避：関節の回転角をバッファに退避する。
・復帰：関節の回転角をバッファから復帰する。
・X軸反転：X軸の回転角を反転する。
・Y軸反転：Y軸の回転角を反転する。
・Z軸反転：Z軸の回転角を反転する。
・初期化：回転角を初期化する。

5 サブメニューの操作方法

a) 基本事項

CGR.Xは2つの構造体編集画面を持っている。

サブメニューには，それぞれの編集画面に対する「ローカルな」操作を集めてある。したがって，左右の編集画面の下に別々に同じメニューバーが並んでおり，片方の操作結果はもちろんもう片方の編集画面には影響しない。

なお，頻繁に使うと思われるサブメニューの機能を，キーボード上に割り振ってある。

キーでサブメニューの操作を行うとき，以下のようなことが有効である。

・キーによる操作の対象となる編集画面は，基本的には，

「マウスカーソルのある側」

である。

・シフトキーを押しながらキーによるサブメニューの操作を行うと，常に左側の編集画面に対する操作となる。

・CTRLキーを押しながらキーによるサブメニューの操作を行った場合，常に右側の編集画面に対する操作となる。

具体的なキー割り当てについては，サブメニューの操作説明それぞれの項の，最後のカッコ内に記しておく。

b) 表示

サブメニューの，「表示」と書かれたメニューバーの上でポップアップメニューを開くことにより，構造体の表示設定の変更を行うことができる。

各項目の概要を列記する。

・初期位置：構造体の表示位置を，初期の位置に戻す（Zキー）。

・前から：構造体を正面から見ることができる（Aキー）。

・右から：構造体を右側から見ることができる（Qキー）。

・上から：構造体を上から見下ろすことができる（Iキー）。

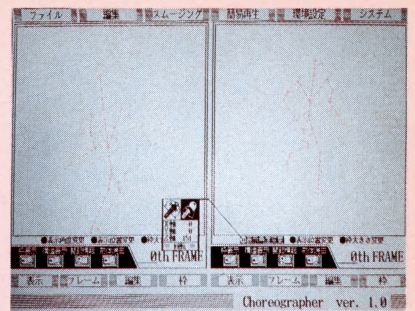
・斜めから：構造体を斜めから見ることができる。

c) フレーム

「フレーム」と書かれたバーの上でポップアップメニューを開くと，以下のような項目が現れる。

・飛び先指定：編集するフレーム番号を，スライドの数値入力により変更する。

・次の標本値：編集するフレームを，次の標本値に移す。標本値が見つからない場合は，フレームは変更されない（Xキー）。

・前の標本値：フレームを直前の標本値に移す（Sキー）。

d) 編集

「編集」と書かれたバーの上でメニューを開くと，左右いずれか目的の編集画面に対して以下のような編集が行える。

・退避：現在の構造体のポーズをバッファに退避する（Cキー）。

・復帰：バッファに退避してあるポーズを編集中のフレームに復帰する（Dキー）。

・標本値開放：編集中のフレームが標本値として設定されている場合，それを解除する。

・コピー：フレーム番号をスライドで指定することにより，編集中のポーズを指定のフレームにコピーする。

・初期化：ポーズを初期化する。ただし，事前に初期化直前のポーズをバッファに退避する。

ポーズのバッファは左右の編集画面とも共通である。したがって一方のポーズをバッファに取り込み，もう一方へ復帰することにより，ポーズをコピーすることができる。

また，誤って初期化をしてしまった場合でも，ポーズの復帰を行うことで初期化前のポーズを回復することができる。

e) 枠

・最大値：「枠」の大きさを，現在のポーズをちょうど包むような大きさに設定する。

当然のことながら，枠表示のスイッチが入っていなければこの操作は画面には反映されない。また，枠の大きさの設定値は左右の編集画面とも別々に用意されている。

6 メインメニューの操作方法

編集中のフレームだけでなく，比較的広範囲に影響を及ぼす操作は，画面上部に並ぶメインメニューに割り当ててある。

a) ファイル

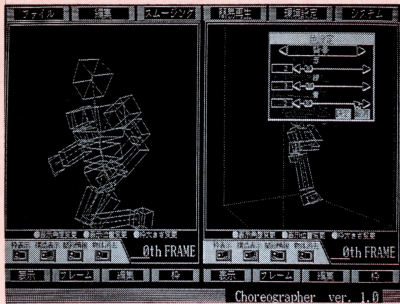
・ロード：CGR.Xで書き出したフレームソースファイルを読み込む。

・セーブ：編集したフレームのポーズをフレームソースとして書き出す。その際，セーブ開始フレームと，終了フレームを指定する。

・MFE形式ロード：MFE.Xで書き出したファイルを読み込む。その際，ファイルを何フレームとして読み込むかを指定する。

ファイル名の選択は，共通のファイラ上で行う。

・目的のファイルがすでに存在する場合，そのファイル名の上でマウスの左ボタンを押して決定ボタンを押せば，ファイルが選択される。

・カレント上にたくさんのファイル，またはディレクトリがある場合，ファイラ右の上下のボタンを押すか，スライドを直接動かせば表示範

囲を移動させることができる。
・ドライブの変更は，ファイラ右上の左右のボタンを押すことで行う。
・新しいファイルを作るときなどは，ターゲットファイルの表示部分を左クリックするか，直接キーボードを叩いて入力する。

b) 編集

サブメニューの編集とは異なり，バッファを使った作業を行う。
・カット：先頭フレーム，最終フレームを指定すると，その間のフレームのポーズすべてをバッファに取り込む。取り込まれた部分は取り除かれ，カットした最終フレーム以降のフレームで埋められる。
・コピー：同じく先頭，最終フレームを指定すると，その間のフレームをバッファに写す。カットと異なり，編集用のフレームの内容は変わらない。
・ペースト：バッファ内のフレームを，編集用のフレームに落としていく。その際，ペーストする先頭フレームを指定する。
一度編集したポーズ群を，よりゆっくり動かしたり，あるいは逆に早く動かすようにしたいときがある。そのようなときには次のような機能が便利である。
・引き伸ばし：バッファ内のフレームを，指定のフレーム内に収まるように加工する。加工後，取り込んだフレームを何フレームに収めるかを指定する。

c) スムージング

・全回転角：編集時に設定された標本値をキーフレームとして，ポーズの中割りをする。開始フレーム，終了フレーム，中割りの滑らかさを指定する。
・ただし，開始フレーム，終了フレームは，ともに標本値でなければならない。
「手を下ろしたポーズ」「振り上げたポーズ」「手を真横に突き出したポーズ」がそれぞれ0，10，20フレーム目にあり，みな標本値として設定されていたとしよう。そこで0から20フレームまでをスムージングすると，手が下・上・横と順番に動く動作を得ることができる。
滑らかさに大きな値を設定すると，次の動作への移行が「滑らかに」なる。0を設定すると，直線的な動きになる。

d) 簡易再生

・順方向：フレームの正方向に簡易再生を行う。指定した再生終了フレームに達すると，若干の「止め」が入る。
・ループ：フレームの正方向に，止めを入れずに連続再生をする。
・逆方向：フレームの負方向に簡易再生を行う。止めが入る。
・逆方向ループ：フレームの負方向に，止めを入れずに逆方向再生を行う。
4つの再生に共通して，次のような値を設定する。
・再生開始：再生を開始するフレーム（逆方向再生の場合は終了フレームとなる）。
・再生終了：再生を終了するフレーム（逆方向再生の場合は開始フレームとなる）。
・1秒間のコマ数：1秒間に再生するコマ数を設定する。1から30まで。
・表示位置：再生の際の表示角度，位置，構造体の表示/非表示などの情報を，編集画面の左右どちらから得るかを設定する。
簡易再生をする直前に，編集画面の片方を骨組みだけの表示にしておき，表示位置でその側を選択すると，再生前の待ち時間がかなり短縮される。

e) 環境設定

さまざまな環境値をカスタマイズすることができる。
・操作環境：操作に関する環境値を設定する。
（物体消去の時間上限）編集中，構造体の描画に時間がかかる場合，自動的に骨組みだけの表示とする。その際のしきい値を設定する。
（リピート開始までの時間）スライドボリュームのボタン操作において，リピートを開始するまでの時間を設定する。
（リピート間隔）ボリュームのボタンのリピートする間隔を設定する。
（マウス移動の反映度）角度，座標値の変更の際，マウスの移動量を値の増減に反映させる度合を設定する。
・色設定：文字，構造体などの色を設定する。
・環境保存：環境値をCGR.ENVという名前のファイルに保存する。
なお，CGR.ENVはCGR.Xと同一のディレクトリに置くことで，起動時に自動的に読み込まれる。ファイルが見つからない場合は，システムの既定値を環境の設定値として使用する。

f) システム

・Human　68kを呼び出す：子プロセスとしてCOMMAND.Xを起動する。EXITでCGR.Xに戻ることができる。
・終了：CGR.Xを終了し，コマンドラインに戻る。

7 構造ファイルの書式について

構造ファイルの書式は，基本的にDōGAのフレームファイルにおける構造体の書式に準ずる。構造体の書式に関しては，DōGAのマニュアルどを参照していただきたい。
ただし，CGR.Xに固有の機能を活用するため，CGR.Xの構造ファイルでは以下のような記述が許される。

a) 構造体の初期設定値

CGR.Xの構造ファイルの先頭において，以下にあげる3種類の値が必ず設定されていなければならない。
・pos(X Y Z)
構造体を表示する初期位置を，X，Y，Zの整数値で指定する。サブメニューで「初期位置」を選ぶと，構造体はこの位置に戻される。
・scale(S)
編集時の構造体のスケールを，整数値Sで指定する。500くらいにとるとよい。
・center(X Y Z)
構造体の表示角度を変更する際，回転の中心になる座標を構造体の中心からの相対座標X，Y，Zで指定する。

b) 構造体ブロック内の設定値

DōGAの構造体では各関節を表す領域をブロックと呼ぶ。また，ある関節があり，その先につながった関節を「子関節」と呼ぼう。
CGR.Xの構造ファイルで子関節を作るにはDōGAのフレームファイルと同様に親関節のブロックから，ちょうどC言語のように中括弧”{　～　}”ではさまれた領域を設け，その中に親関節からの相対位置などを記す。
ブロックの中では，以下のようにフレームファイルと同様の記述が許される。
・move(X Y Z)
親関節からの相対位置を座標X，Y，Zで指定する。
・obj ファイル名
そのブロックの物体名を指定する。
・rotx(R1) roty(R2) rotz(R3)
回転角の初期値をR1，R2，R3で指定する。特別な場合以外は指定しなくてよい。
また，CGR.X専用の設定値として以下のようなものもある。
・name(名前)
ブロックの名前を指定する。
・rmsk(M)
軸マスクを設定する。値Mを2進化し，下位ビットからそれぞれX，Y，Z軸が割り振られており，ビットが立っていた場合その回転角を変更できない。特別な場合以外は指定しなくてよい。
・vir(N1 N2 N3)
MFE.Xで用いられている，関節の回転角の変数名を指定する。これを指定しないと，MFE.Xで作成されたファイルを読み込むことができない。

c) 形状ファイルについて

構造ファイルで指定された形状ファイルは，次のような手順で読み込まれる。
・まず，構造ファイルの拡張子を.SUFに置き換えたファイルを読み込む。複数の形状がひとつになったファイルであれば，物体名と構造ファイル中の指定名とを突き合せ形状を得る。
・同一名の形状ファイルが見つからない場合，または同一名のファイル中に必要な形状がすべて揃っていない場合は，必要な形状をファイル名とするファイルを読み込む。
必要な形状がすべて揃わなければ起動は失敗に終わる。
ファイル管理が煩雑になるのを避けるため，できるだけ構造ファイルと同一名の形状ファイルに必要な形状をまとめておくほうがよい。
また，編集に用いる形状は少々複雑なものでもかまわないが，当然複雑さに応じて描画速度も遅くなる。適当にシェイプアップした形状を使うことをお勧めする。

8 最後に

CGR.Xはフリーソフトである。プログラム本体，またはソースなどについて，商目的でない限り，使用に制限は一切設けない。

お姉さんが教えてあげる

CHERRY BOY

Hamazaki Masaya 浜崎 正哉

**1993年6月号で発表されたパチスロゲームがバージョンアップしました
ドラムは大きく，台のパネルも一新され，よりそれらしく仕上がっています
パチスロの雰囲気をつかみつつ，軽い気持ちで遊んでみてください**

1993年6月号で発表したパチスロエミュレータをちょこっといじり，名前を「CHERRY BOY」と変えて装いも新たにバージョンアップしました。今回のバージョンアップで変わった点は，チャチな画面を一新したのと連チャンシステムを搭載したところです。基本的なアルゴリズムは，6月号で発表したものとまったく同じ。ドラム半周スベリもそのまんまとなっています。

また，ちょっといい加減な連チャンシステムにより，いままでも負け知らずの出玉がさらに爆発しています。特に設定5だとムチャクチャ出ます。とりあえず，日頃負け続けてうっぷんが溜まっている人のストレス解消に役立つことでしょう。ほどほどにしておかないと，現実とのギャップに苦しんだりするかもしれませんけどね。

起動方法

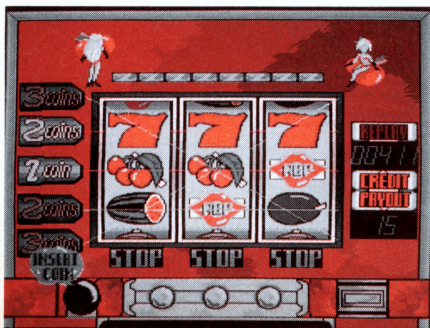
まず，起動方法です。Z-MUSICと音色データを，

A>ZMUSIC -SSLOT_SND.ZMS

として組み込み，

A>SLOT

で，起動します。起動時には，

CHERRY.SPD
CHERRY.BG
CHERRY.PAL
その他AD PCMデータ

を同じディレクトリに置くようにしてください。ファイルが揃っていない場合でも，強引に起動しますので注意しましょう。

777が揃う瞬間！

遊び方

ゲームを起動すると，ウインクしたパッキンのお姉さん（と思ってくれ）とともにメニューが表示されます。メニューは，

・GAME START……ゲーム開始
・CONFIGRATION……機械の各種設定
・END……ゲーム終了

となっていて，CONFIGRATIONでは，以下の設定を行えます。

・設定……機械の設定（値が大きくなるとよく出る）
・打ち込み……自動，手動モードの切り替え
・回数……自動モードでの打ち込み回数の設定
・連チャン……連チャンシステムを組み込むかどうかの設定
・スピード……ドラム回転スピードの設定
・モーニング……ゲーム開始時にBIG BONUSのフラグを立てるかどうか設定

そして，ゲーム中のキー操作は，

・E……ゲーム終了
・C……クレジット投入
・スペース……ドラムスタート
・7,8,9……ドラム停止

となっています。ゲーム終了後には，前回のバージョンと同じく，揃った役の回数とコイン獲得グラフが表示されます。

プログラムは？

新たに加わった連チャンシステムですが，あまりたいしたことはしていません。まず，BIG BONUS(777)を当ててから，そのボーナスゲームが終わったときに，そのあと何連チャンするか確率表に従って抽選します。そして，そのあとの50ゲームは，その連チャン回数のボーナスを消化するまで，BIG BONUSの確率が30%にアップするようにしているのです。

話は変わって，このプログラムでは，新たにX-BASIC用の外部関数を追加しています。それは，

・v_disp()……垂直帰線期間を検出するまでループする
・pri_chg()……半透明機能の設定とテキスト，スプライト，グラフィックのプライオリティを変更する

の2つで，ともに引数，返り値なし，となっています。そして，コンパイルするためには，Z-MUSIC用のライブラリも必要とします。これは，今回の付録ディスクには収録されていません（Oh!X Books「Z-MUSICシステムver.2.0」に収録予定）。なくてもなんとかコンパイルできると思いますが，音楽，効果音が正常に演奏されませんので，注意してください。プログラムのコンパイルについては，バッチファイルを参照するといいでしょう。

最後に

最近の僕は，山佐の「ニューパルサー」にハマり，たまに勝ちほとんど負け状態の生活を送っています。おかげで財布がピーピー泣いています。でも，面白いんですよね。これといった作為的なシステムなしに(本当はあるのかな？)2,000〜3,000枚コースの波に乗ったときには，ものすごく嬉しいものがあります。

今回の「CHERRY BOY」では，外見にこだわりすぎて中身のほうがわりとおざなりなものになってしまいました。今度こそは，もっと自分好みの面白いパチスロを作ってみたいですね（せめて，負け分を原稿料で稼がなきゃ生活が苦しい……これが本音？）。ぼちぼちとやりますか。

PENJANG!をバージョンアップ

ペンギンと愉快な仲間たち

Asakura Yuji 朝倉 祐二

**ペンギンのキャラクタたちが牌になったかわいい簡易麻雀ゲーム
今回のバージョンアップでライバルも登場するし，役も増えて楽しさ倍増
さあ，君もPENJANG!で優勝してモテモテ男になろう**

おきらくにPENJANG!を遊ぼう

ご無沙汰しています。最近は身の周りが忙しくて，なかなか自分の自由な時間がとれない朝倉です。それでもなんとかPENJANG!をバージョンアップすることができましたので，発表したいと思います。

PENJANG!は，1993年6月号「確率遊技シミュレーション」の中でパチスロ，競輪シミュレーションゲームとともに発表された簡易麻雀ゲームです。作りやすい役と作りにくい役の境界線がほとんどなく，また上がり役が5種類と少なかったものですから，結果としてほとんどの上がり役が麻雀でいう「立直のみ」(PENJANG!では「とりあえず」という役名ですが）になってました。もしも上がり役のできやすさの確率を考えたとしたら「とりあえず」で7割を超えてしまうのではないかと思えるほどです。確率という点ではあまりに1点に片寄ってしまい，つまらなかったことと思います。

今回は特にその点を反省してプログラミングしましたので，皆さんに少しは楽しんでいただけるゲームになったと思うのですが，さてどんなものでしょうか？

PENJANG!の遊び方

PENJANG!は麻雀やセブンブリッジを知っている方なら，非常に馴染みやすいルールになっています。PENJANG!の基本的な遊び方を説明しましょう。

●キャラクタ選択

ゲームが始まるとまずキャラクタセレクト画面になります。ここでマウスカーソルを動かして自分の分身となるキャラクタを1人と好みの対戦相手を3人選んでください。すでに選択されているキャラクタには赤い枠が表示され，再選択できませんので注意してください。

●ゲームの進め方

キャラクタを選択するとゲーム画面が表示されます。PENJANG!は青，赤，緑の各色について1～9の数字が4枚ずつある合計108枚の牌を使って遊びます。

まず4人のプレイヤーに各々8枚づつ計32枚の牌が配られます。画面一番下に横1列に表示されている8枚の牌がプレイヤーの手牌です。それから残りの76枚の牌を親から順番に「1枚ツモって（注1）は1枚捨てる」という処理を誰かが上がり役を完成させるか，ツモる牌がなくなるまで繰り返してゲームを進めます。

ツモ処理はコンピュータが自動的に行いますので，プレイヤーの番になると牌をツモってきたところで捨て牌の選択処理になります。ここで上がり役（後述）に関係のなさそうな不要な牌にマウスカーソルを重ねて左ボタンをクリックすると捨て牌処理がされます。

またここで，右ボタンをドラッグしますとメニューが表示されます。立直（注2）をかける場合は，ここで右ボタンをドラッグしたままマウスカーソルを「りーち」に合わせて右ボタンから指を離してください。そのあと捨て牌を決定しますと，自動的に立直がかかります。一応フリテン（注3）チェックもしています。上がり牌が出たときには，「でたぁ」と表示され，ゲームの進行が中断しますので左ボタンをクリックしてください。

なお前回発表したPENJANG!は，必ず立直をしないと上がれない仕様になっていましたが，今回からはツモの場合に限って立直なしで上がれるようになっています。ツモをキャンセルするかどうかは，画面に指示が出ますのでそれに従ってください。

親が上がった場合には，すべての得点が倍になります。親以外の人が上がった場合には，親が変わります。全員が2回親をやるとゲーム終了です。成績が表示されますので，自分の順位を確認してください。2位以内なら2回戦進出です。3回戦でトップの得点を取ると優勝となり，ゲーム終了です。

注1：ツモ

牌を1枚持ってくる動作のこと。このとき上がればそれをツモ上がり（麻雀ではツモと宣言します）といいます。

注2：（立直）りーち

立直はあと1枚で上がり役が完成するというときに宣言するものです。立直は他人の捨てた牌を自分の牌に加えて上がり役とすることができますが，立直を宣言したあとに手牌を交換することはできなくなります。PENJANG!では立直のあとは自動的に捨て牌を決定するようになっています。

また，立直を宣言するときには立直代と

動物たちの熱い勝負が展開されている

して10点を支払います。払われた立直代は，すべて上がった人にまとめて渡されます。

もしも間違えて，あと１枚で上がれない場面で立直をかけてしまい，しかも誰も上がることができないとチョンボとしてすべてのプレイヤーに20点を支払うことになりますので注意してください。この場合は，誰かが上がればチョンボは見逃されます。

こんな役で上がっていては男がすたる

やっぱり男だったらいっしょく狙いだね

注３：フリテン

立直前にすでに自分が捨てた牌で他人からロン上がりすることは禁止されています。これをフリテンといいます。この場合はツモった牌でしか上がることができません。

注４：ロン

他人の捨て牌で上がり役が完成すること。

上がり役紹介

PENJANG!では同色の連続する数字が３枚集まると順子(しゅんつ)，または同色の同じ数字が３枚集まると刻子（こうつ）として扱います。

順子の例　　　345，123，678

刻子の例　　　111，222，333

手牌の中で順子と刻子の合計が３つにすることが目的です。順子だけ，刻子だけというのも可能です。この組み合わせにも難易度があって，確率的に簡単なものは低く，難しいものには高い得点がつけられています。PENJANG!には，以下の11種類の上がり役があります。

●３色同順　　50点

青，赤，緑の３色を使っての同順の順子が３つあるとき

●３色同刻　　50点

青，赤，緑の３色を使っての同じ数字の刻子が３つあるとき

●さんしょく　　15点

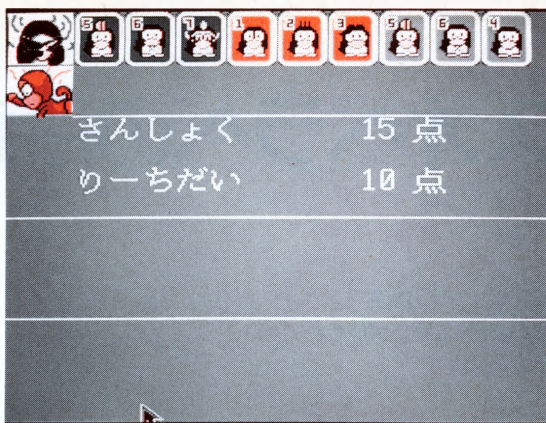
青，赤，緑の３色を使っているとき。３色同順，３色同刻が成立しているときは認められない

●いっしょく　　30点

青，赤，緑のどれか１色で完成されているとき

●さんあんこ　　60点

刻子だけで上がったとき

●たんやお　　10点

２～８の牌だけを使っているとき

●ちゃんた　　30点

たんやおの逆で，すべての順子，刻子が１か９を含んでいるとき

●いーぺーこー　　40点

同色で同順の順子が２つあるとき。さんあんこが成立しているときは成り立たない

●いっぱつ！　　10点

立直の直後に自分でツモって上がるか，もしくはツモまでにロン上がりしたとき

●いっつー　　60点

青，赤，緑のどれか１色で，１～９のすべての牌を使用しているとき

●とりあえず　　５点

上記の上がり役のどの条件にも当てはまらないが，とりあえず順子と刻子の合計が３つあるとき

コンピュータの思考について

コンピュータは「早上がり」，「たんやお」，「さんしょく」のいずれかを狙うようにプログラミングされています。つまり「いっしょく」や「いっつー」などをコンピュータが完成したとしたら，それは本当に偶然のことなのです。いうまでもありませんが，牌のすり替えなどのインチキはいっさいしていません。

攻め方の傾向はPENJANG.BASの380～450行で0～100の範囲で定義します。数字が大きいほど早上がりの攻め方になっていきます。ただし早上がりですので，上がり役の得点はそれほど高いものにならないことが多いです。逆に数字が小さいと高い役で上がってくる確率が高くなります。

リコンパイルについて

プログラムを改造してみたいという方や，バグを見つけて自分で直したいという心のやさしい方にコンパイル方法を説明します。GCCを使える環境が構築されているならアーカイブに含まれている“tcompile.bat”というファイルを実行すると自動的に実行ファイルを作成してくれます。バッチファイルを実行する前に，カレントディレクトリへ“gra_l.o”をコピーしておくことを忘れないようにしてください（これはパチスロゲームCHERRY BOYのディレクトリに収録されています)。

最後に注意事項&お願い

PENJANG!には西川善司作曲のBGMが使われています。そのためPENJANG!を実行する前に，必ずPCM8.X,ZMUSIC.Xを常駐させておいてください。

もしもスプライトが定義されなかったりちゃんとPCM8.X，ZMUSIC.Xを常駐させているのに音楽がおかしいといったときは，CONFIG.SYSのFILES=の値を15くらいに書き直してからリセットして再実行してみてください。

ところでバグがないといい切る自信はまったくありません。もしもバグを発見したり，質問事項などがありましたらOh!Xの「PENJANG!で質問」係宛に手紙を送ってください。もしくは，通信をやっていてZ-MUSICのサポートネットにアクセスできるのなら，ID番号MIYA0350が私の知り合いですので，そちら宛にメールで連絡してくれても結構です。できるかぎり対応したいと思います。

さてさて，駆け足でPENJANG!についていろいろ説明してきました。これを読んで興味をもった方は，ぜひとも遊んでみて優勝を目指してがんばってください。コツをつかむまで，結構たいへんですよ。

またいつか誌面でお会いできるといいですね。

SLASHが秘める無限の可能性

壮大なるスペースオペラへの序章

Yamada Junji 山田 純二

「SION」シリーズもワイヤーフレームの世界からポリゴンの世界へ
ポリゴナイザライブラリ「SLASH」の実力を実感できるサンプルです
シェーディング処理されたポリゴンを堪能してください

横内氏制作のポリゴナイザライブラリ「SLASH」に触る機会があったので，ちょっと試してみました。すると不思議なことに，あれよあれよといろいろなルーチンが加わり，ついには「SION IV」のデモプログラムとなってしまいましたとさ。これが，今回のサンプルプログラムの制作背景です。

う～ん，それにしてもすごい。はっきりいって，これほどまともにポリゴンを扱える環境を手にできるとは，思ってもみませんでした。横内氏は記事中で「それなりに高速……」などと控え目な表現をしていますが，実際に使ってみた感想は，「なんでこんなものが動くんだ」とただただ驚愕するばかり。

とりあえず，デモバージョンということですが，狂喜乱舞するポリゴンたちをゆっくりご観賞ください。

使い方

付録ディスクのバッチファイルによって，作成されるディスクをそのまま立ち上げてもらえば自動的に起動します。解凍作業を見ればわかるとおり，このデモ版ではスイッチの指定により，効果音なしモードと「SION II」の効果音を流用するモードの2種類の起動方法があります。

効果音なしモードのときは，

A>DEMO_S4

として起動するだけでOKです。効果音使用モードのときは，まず，「SION II」で使われているAD PCMデータと音色データをZ-MUSICに，

A>ZMUSIC -N -BSION2 -SNEIRO.ZMS

のようにして組み込んでください。そのあと，メインプログラムを，

A>DEMO_S4 -M

のように“-M”スイッチをつけて起動すれば，レーザーと敵爆発音が鳴るようになります。もしも，BGMをつけたいと思っている人は，メインプログラムを実行する前に曲データをOPMにコピーしておいてください。解凍されたディスクには，PCM8.Xが自動的に組み込まれるようになっていますが，10MHz機のユーザーは外しておいたほうがいいでしょう（動くことは動くけどね）。このあたりの設定は，各自バッチファイルを書き換えるなどして対応してください。

ゲーム中のキー操作は，2,4,6,8キーで上下左右の移動，Zキーでレーザーの発射，F10キーで終了します。なお，ゲームを終了するときには，必ずF10キーを押して終了するようにしてください。これは，プログラム中で垂直帰線期間割り込みを使用しているためです。インタラプトボタンなどで強制的にプログラムを終了してしまうと，割り込み制御がそのまま残ってしまい，ほぼ確実に暴走します。

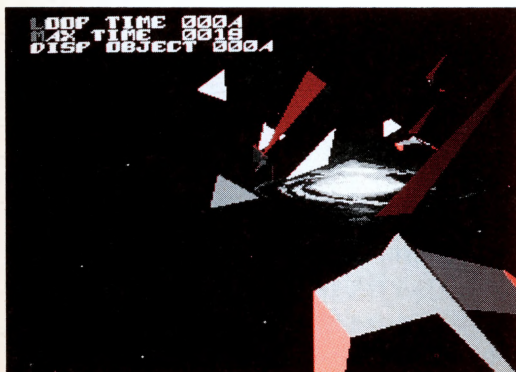
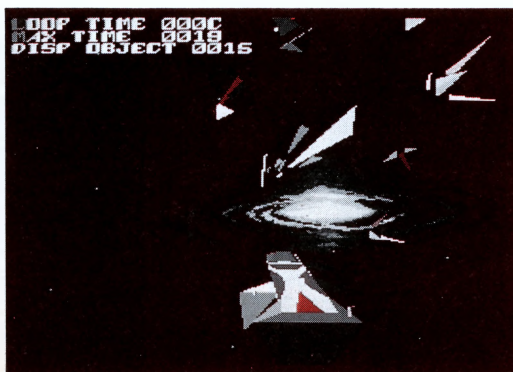
プログラム内部について

なにぶん，未完成ということもありいろいろと実験的なルーチンが随所に埋もれています。解析をしようとした人にはもうしわけありませんが，非常にわかりづらいプログラムとなってしまいました。

つれづれなるままにエディタへ向かいプログラムを作成したわけですが，一応，速い機種では滑らかに，処理が重くなったら描画コマ数を落とす，という基本方針で開発を進めました。つまり，X68030だろうが初代X68000だろうが同じ速度で動くことを前提としたのです。

結論からいって，この処理はおおむね成功しています。極端に処理が重くなったり軽くなったりすると，体で感じてしまいますが，たいていの場合は，それほど違和感がないはずです。

では，基本的なアルゴリズムを説明していきましょう。まず，どのようにして時間を計測するか考えます。最初に考えつくのは，タイマを使っての時間計測です。いちばん正確で細かな計測が行えますが，個人的に余計に割り込みのかかるのは嫌だし，まったく知らない領域でもあるため，まずは却下。

次に考えられるのが，垂直帰線期間割り込みを使う方法です。これは，同じ割り込みでも手軽に使える利点がありますが，皆さんご承知のとおり，垂直帰線期間開始から次の垂直帰線期間開始まで1/60秒もかかります。これでは，計測間隔が粗すぎると考え，この方法も却下。

シェーディング処理が美しい

ちょっと大きな戦艦も登場

最終的には，X68000が起動してからの経過時間を返す，IOCSコール$7F_H$(_ONTIME)を使用することにしました。これは，X68000のIPL ROMが起動してから1/100秒単位で時間計測をするものです。IOCSコールは遅いという話もありますが，まあ，扱いが簡単だし1/100秒単位ならば，実用上問題はないでしょう，という結構安易な考えで採用しています。

さて，かかった時間に対して描画コマ数をどうやって落とすか，ということですが，これは，1ループにdt秒の時間がかかったから，距離(dx)だけキャラクターたちを動かしてやることで実現できます。

つまり，メインループが1ループするたびにある一定の移動を行わせるのではなく，メインループが1ループするまでの時間を計測し，その経過時間(dt)に速度(v)をかけて求められる距離(dx)だけを移動させるのです。いわゆる運動方程式，

dx＝dt×v……(1)

で経過時間に対する移動すべき距離が算出されます。しかし，速度が十分な大きさであればなんの問題もないのですが，ここで，キャラクターを非常にゆっくり動かしたい場合はどうしたらいいのでしょうか。当然，速度は小さくなり小数の値を取ることになります。はい，マシン語レベルでは実数を扱うのが面倒臭いですね。

この問題は，速度をどのようにして求められるかを思い出せば解決します。速度(v)は，移動距離(X)割る移動時間(T)，

v＝X/T　余り……AM

という式で求められます。これを式1に代入してやると，

dx＝dt×(X/T)

になり，さらに速度を求める段階で生じた余り(AM)をXに足してやることで，

dx＝dt×((X＋AM)/T)

が導き出されます。そして，速度を求める段階で生じた余りは，次の演算のために保存しておく必要があります。あとは，X,Y,Z軸のそれぞれに計算してやれば，3次元の移動ルーチンの出来上がりです（図1）。

また，よりリアルな動きを求めるのであれば，等速度運動だけでなく加速度運動もサポートしなくてはならないでしょう。これは，今後の課題としておきます。

さて，実はこの時間という概念を導入すると，キャラクターが起こすすべての行動に時間を考慮しなくてはなりません（当たり前）。いままでは，単純にカウンタを導入して順次カウントダウンを行い，定められた値で行動を起こせばよかったので，非常に簡単にルーチンを作成できました。しかし，キャラクターの出現個数や描画にかかる時間など，さまざまな要因によって経過時間が変わってきます。面倒臭いのは，この経過時間がまちまちとなる点なのです。キャラクターのアニメーションで，

ANI1(nt＝2)，ANI2(nt＝3)，ANI3(nt＝2)……

(nt：次のアニメーションまでの経過時間)というようなタイミングで，アニメーション動作を行わせようとします。経過時間が十分に小さければ，ANI1～ANI3までを順次実行していけばいいのですが，処理が重たくなったとき，ntの値が半端な値を取ることになります。ひどいときになると経過時間が大きすぎて，ANI1から一気にANI3まで飛ぶこともあります。

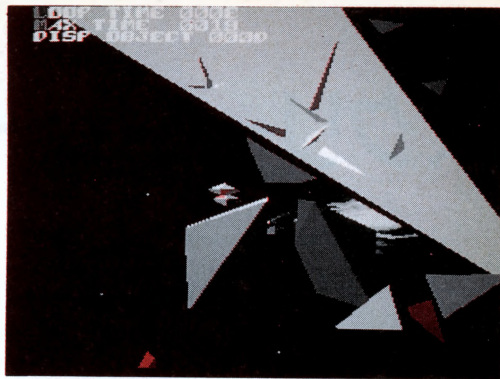
さらに，移動中にこういった事態が起きると話がもっとややこしくなるのです。A点から始まって，B,C点を経由しD点まで移動を行う場合を考えます。すると，最初の移動目的地であるB点まで移動するときに，必ずといっていいほどB点までの移動時間と経過時間の総計が一致しません。つまり，B点を経由するためにB点に到達する経過時間を求めて，その移動量を計算し，そして，B点までの移動時間と経過時間の総計の差を，新しくB点からC点まで移動するための経過時間として，移動量を計算しなくてはならないのです。

さらに経過時間が大きくなると，今度はA点からB点を越え，さらに先のC点の先まで一気に移動する場合もあります。この場合も同様に移動量を計算する必要があります。ちなみに今回のルーチンでは，このようなことは起きない，という前提で処理をはしょっています。

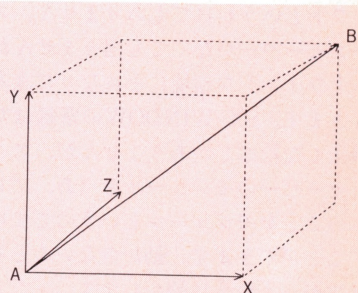
キャラクターがでかくなってもちゃんと動く

これからの展望

一応，「SION IV」のデモンストレーションプログラムですが，自機を操作できるとはいえゲームとしては成立していません。自機に対する当たり判定はないし，キャラクターの出現パターンは単にデータがループしているだけ。出現パターンデータもかなりいい加減に設定しています。実質上は「SLASH」のデモンストレーションと考えてもらったほうがいいかもしれません。

それでも，これからプログラム中で使われる基本ルーチンの構造を把握することができたし(自分なりにね)，結構見栄えのするデモとしてインパクトはあると思っています。今度こそは，皆さんの期待を裏切りたくないし，自分自身にけじめをつける意味でも全力を尽くすつもりです。

今回やっていることは，算数レベルのアルゴリズムに小学生以下のコーディングと，解説するのも恥ずかしいくらいですが，それでもこれだけのものを動かすことができるのです。もしも，1クロックの無駄もなく最適化されたルーチンを使ったら，どれだけのものができるか僕には想像ができません。ポリゴンという表現上の制約がありますが，そんなものは技術，感性，根性そして努力しだいでなんとでもなるはず（甘いかな？）。

制作はこれからが正念場。もっともっと熱くならなくては……。

図1

経過時間の間の移動量は，AからBまでの移動時間をTとして，X，Y，Z軸それぞれ，

$$Vx=\frac{X+AM}{T}$$

$$dx=dt\times Vx$$

$$Vy=\frac{Y+AM}{T}$$

$$dy=dt\times Vy$$

$$Vz=\frac{Z+AM}{T}$$

$$dz=dt\times Vz$$

で求められる。

スクリーンセーバー画面暗前

Ishigami Tatsuya 石上 達也

焼き付きから画面を守るスクリーンセーバー
以前予告していたとおり，ついに石上版の登場です
SW-WINDOWの画面をもっともっと華やかにしてみませんか

スクリーンセーバー

シャープから発売されているディスクアクセサリ集の中にも純正のスクリーンセーバーが収められているのですが，それとは別に，5月号で予告した石上版スクリーンセーバーを発表します。

同じ画像をCRTに表示したままにしておくと「焼き付き」現象を起こしてしまいます。そこで，コンピュータがなにも操作されていないときに，無理やり画像を変えてやるためのソフトウェアが，スクリーンセーバーです。

しかし，5月号でもちょこっと書いたように，パソコンのCRTは，ちょっとやそっとじゃ焼き付きませんので，必ずしも必要なソフトというわけでもありません。スクリーンセーバーがなくても困りませんが，あるとウィンドウ環境にそれなりに深みが出てきます。SX-WINDOWの壁紙はただの静止した絵でしかありませんが，スクリーンセーバーのモジュールを（あ，プログラムのことをモジュールというのです），動かすことができます。動かさないこともできますが，ふつうは動かします。

そういうわけで，壁紙よりはやや自由度の高いインテリアみたいなものです。画面の焼き付き防止のため，という目的以外にも，煮詰ったときの刺激によい，とかなんとか理由をつけることもできますが，かわいいから，とかいう理由で部屋の片隅に置かれたぬいぐるみみたいなもんです（註：私の部屋にぬいぐるみはありません）。

FISH!

私の作ったサンプルのモジュールでもしばらくは遊べるかもしれませんが，すぐに飽きるでしょう。

で，すぐには飽きないモジュールを，と，ストライダー横内氏に依頼したのがFISH.Xです。彼がアニメーションパターンおよび背景の作成を行い，私がそれらをスクリーンセーバーのモジュールとしてまとめました。

なにはともあれ，まず使ってみてください。

1）　SX-WINDOWを立ち上げる
2）　SAVER.Xをダブルクリックで起動する
3）　すると，画面暗前という名前のウィンドウが開くので，そのウィンドウの中へFISH.Xのアイコンをドラッグして放り込んでやる
4）　ウィンドウ上にいろいろとボタンが出るが，とりあえず，右下にあるDEMOと書かれたボタンを押す

FISH以外のモジュールと画面暗前本体は私が作成しました。FISH以外のモジュールは，モジュールの作り方のサンプルプログラムみたいなものです。作った当時は，けっこう使えるものだと思ったのですが，FISHを見てから考えが変わりました。サンプルプログラムということになってしまったかわいそうなプログラムたちです。

モジュールの書き方は，来月号で紹介しますので，これらの解説はそのとき行います。

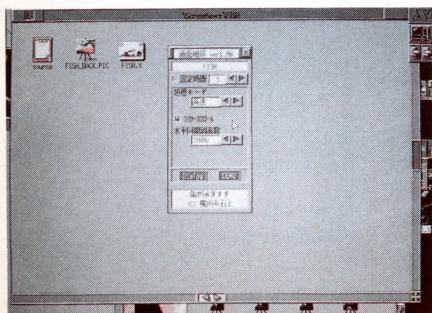
さて，FISH.XはSX-WINDOWのプログラミングなんてやったことのない横内氏と68000のマシン語なんてほとんどわからない私の合作です。書いていて思い出しましたが，私はスプライトとかBGというのもよくわかりません。

石「あの，テキスト画面の一部が乱れるんですけど」
横「あ，回転ルーチンでテキスト画面をワークエリアに使っているんです。それで画面が乱れるんです」
石「ふつう，ワークエリアってメインメモリ上にとりません？」
横「ビットマスクを使ってるんでテキストじゃないとまずいんですよ。プログラムで処理を振り分けるとスピードが落ちるし」
石「……そういうものなんですか」
横「終了時に，テキスト画面を書き直させるというのは，どうですか？」
石「え？　しかし……」
横「じゃ，乱れる部分をあらかじめ退避するようにしましょう。終了時にそれを復活させれば，どこにも問題はないですよね」
石「……。はい」

というようなモジュールです。

SX-WINDOWの作法などということは露ほどにも考えられていません。水の揺らぎはラスタースクロールを使って表現していますし，FISH.Xでの処理はほとんどスーパーバイザモードで行っています。

「動くコードはすべて美しい」

真実とはいつも驚くほど短い言葉に表さ

スクリーンセーバー本体

れているもんです。

それでも，文句がある人は，直接，横内氏まで，どうぞ。ちなみに氏は大学の殺人拳法研究会に出入りしているそうで，最近抜き手が甘くなったとおっしゃっていました。今頃はきっと，熱した砂を相手に，「てやっ，てやっ」と練習に励んでいるところですから，文句をつけに行く人は練習のジャマにならないようにしてください。

そんなわけで，このモジュールには，ほかのSXプログラムには見られないような設定がいくつかあります。

●動作モード

SX-WINDOWの動作スピードというのは，ほかのタスクとの兼ね合いで決まるので一律には決まりません。多くのタスクを動作させれば，全体のスピードは低下しますし，動作中のタスクを減らせば，それだけ処理スピードを上げることができます。

FISHでも魚の発生タイミングや移動速度やらの定数は処理スピードに依存します。これらの定数は，最初にコマンドライン上でFISHを開発したこともあって，ほかのタスクがいっさい動作していない状態で本来の動作をするように作られています。

しかし，そのために，ウィンドウを片っ端から閉じたり開いていったりするのはあまり賢明なことではありません。

そこで，この動作モード選択用のスイッチを設けました。このスイッチが高速モードに入ってると，FISH.Xは，キーボードが押されたり，マウスがいじられたりしなければほかのタスクへ処理を移さないようになります（スクリーンセーバーの終了条件と同じです。画面暗前のやっていることの一部をFISH.X内でも行っています）。

●ラスタースクロール

先ほども述べましたが，水の揺らぎを表現するのに，ラスタースクロールを使って背景のグラフィック画面を揺らしています（自分で書いてるくせにいうのもなんですが，私はラスタースクロールのなんたるかをまったく理解していません）。

SXでラスタースクロールを使うようなアプリケーションが許可されているという話は聞いたことがありません。しかし，裏を返せば，ラスタースクロールを使用するようなSXプログラムはそうそうないわけで，複数のプログラムからラスタースクロールを使おうとしなければ，なんら問題はないわけで，きっと問題はないよなあ，ないといいなあ，ええ，ありませんとも。

もし，万が一にも問題があったときは，このスイッチを切って，ラスタースクロールを禁止してください。

また，SX-WINDOWとの絡みではないのですが，相性の悪いデバイスドライバを組み込んでいると，このラスタースクロールが「よどみ」に見えないで，「ぶれ」に見えてしまうことがあります。

現在，相性の悪いデバイスドライバとして，OPMDRV2が見つかっています。

詳しいことはわかりませんが，タイマ割り込み関係でぶつかっているようです。Z-MUSICを組み込んでいる分には問題はないので，音源ドライバにはZ-MUSICを使用するようにしてください。

もしOPMDRV2でないと不都合が生じるとか，その他の相性の悪いデバイスドライバが切り外せないというときには，とりあえず，このラスタースクロールスイッチを切っておいてください。水の揺らぎは再現されなくなりますが，割り込み関係でぶつかることはなくなると思います。

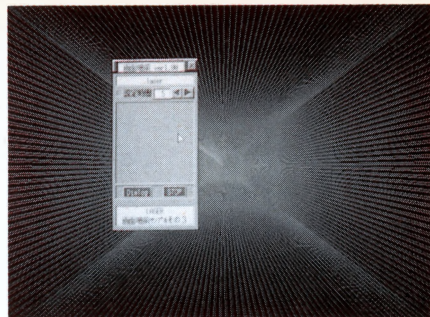
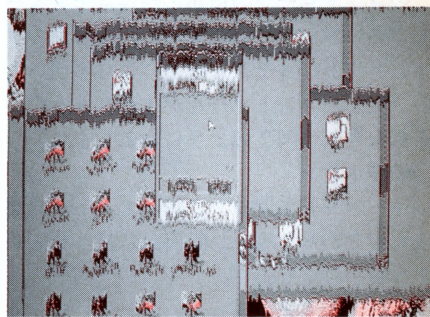
●水平同期周波数

ここまでやったら，やっていけないことは残っていません。もうディスプレイの水平同期周波数だって変更します。このスイッチを31kHzに設定しておけば，SX-WINDOWから水平同期周波数の変更なしにモジュールの実行を行えます（切り替え時の，カチッという音がしない）。15kHzに設定しておけば，CRTをフルに使った迫力あるグラフィック画面を楽しむことができます（またまた，詳しいことはわかりませんが，横内氏によると15kHz時には，垂直帰線時間が若干短くなるそうで，10MHz機では，一部のキャラクタがチラついてしまうそうです）。

FISHの実行に関する注意点

SX-WINDOWには，ダイアログを管理するにはダイアログマネージャ，テキストを管理するにはテキストマネージャ，というように専用のマネージャが用意されていて，それらをちゃんと使っているうちは，動作が保証されています。

ところが，SX-WINDOW上には，スプライトを管理するスプライトマネージャがありませんので，FISH.Xでは直接ハードウェアを操作しています。つまり，スプライトをいじるようなほかのソフトウェアとの共存はできません。

また，グラフィックを管理するグラフィックマネージャが，ver.3.0から大幅に変更になったようで（私の推測ですが，カラー画像が変則的に扱えるようになった関係上，

サンプルのレーザー

画面を溶かすメルト

完全な上位互換性はないようです。グラフィックマネージャの上位マネージャであるウィンドウマネージャからも，WI_STDG系のウィンドウを開くことができなくなりました。このことによって，グラフィックウィンドウのアップデートに一部不都合が生じますが，今回は修正を見送ります。ver.3対応の開発キットを入手しだい，修正することにします），SX-WINDOW　ver.2とver.3の両方で実行できるように，グラフィックの描画および，パレットの設定は直接ハードウェアを叩いています。グラフィック画面を使うようなアプリケーションとの共存も難しいと思います。

そんなわけで，今回に限り，SX-WINDOWの画面焼き付きを防止するためにFISH.Xがあったのではなく，FISH.Xを実行するためにたまたまSX-WINDOWがあったと思ってください。

スクリーンセーバーの使い方

スクリーンセーバーとモジュールの関係は図1のようになっています。

ユーザーが直接モジュールに対して命令を与えるということは，まずありません。すべての命令は，スクリーンセーバーを通して，モジュールに送られます。というか，送られるように作ってください。

まず，SAVER.Xを起動したときに「画面暗前」という名前のウィンドウが開きます。これがスクリーンセーバーのシェルと呼ばれる部分です。

このウィンドウに，ちゃんと画面暗前に対応した実行プログラムのアイコンを放り込みます。

すると，ウィンドウ上方にそのプログラムの名前が表示され，(そうなるように作ってあるモジュールは) ウィンドウ下方になにかコメントが表示されます。また，なにかの設定が必要なモジュールでは，画面暗前のウィンドウ中央部に，いくつかのコントロールが表示されます（最大4つまで)。このコントロールの種類と内容はモジュールごとに異なっていますが，画面暗前のウィンドウ上にはモジュールによらず使い方の一定したコントロールが4つあります。それらを仮に図2のように命名します。

○動作ボタン

キーボードとマウスに対して，一定時間入力がなかった場合，画面を暗転させるかどうかを決定します。

このスイッチが入っているときにかぎり，暗転を行います。

○時間設定ボタン

動作ボタンが入っていた場合，最後のキーボードあるいはマウスの入力に対して，何分後に画面を暗転させるかの設定を行います。1分から60分まで1分刻みで設定できます。

あまり，この値を小さくしておくと，ちょっと考えごとをするたびに，画面が暗転してしまいます。エディタでプログラムなどを組んでいるときなど，あまり画面がチカチカ切り替ると，たいへんうっとうしい思いをします。

また，SX-WINDOWを立ち上げたまま，1時間近く，ぼーっ，としている人もいないでしょう。もちろん，この場合キーボードやマウスに触れないで，ということですから，ゲームをやってたりするわけではありません。本当に，ぼーっ，とです。

普通の人は，こんなに長い間，ぼけぇ〜〜，っとしていることはないわけで，あまり大きな値を設定しておくと，けっきょく動作ボタンを入れていないのと同じことになってしまいます。

そこらへんをふまえつつ，この値を決めてください。

○DIALOGボタン

モジュールのクレジットなどを表示します。とはいっても，シェルはモジュールに対しダイアログ表示の命令をタスク間通信で送るだけです。どのようなものが表示されるかはモジュールによります。

○DEMOボタン

ふつう暗転は，キーボードとマウスに対して，一定時間操作が行われなかったときに行われます。ということは，ふつうに暗転をさせるには，キーボードとマウスに触れずにしばらく，じっと待っていなければならないわけです。

しかし，なにもせずにじっと最低1分間もコンピュータの前に座っているのは，マヌケです。たとえば，自作モジュールのデバッグ時のように，本来の焼き付き防止以外の目的で暗転を行わせる場合に，このボタンを使用します。

また，モジュールの作り方によっては，「画面暗前」上のスイッチの状態が，すぐに反映されるようにもできますので，対話的な状況で，モジュールの種々の設定ができるようになります。

たとえば，サンプルプログラムのQUIXでは，対話的に線の色を決めることができます。対話的に，というと問題があるかもしれませんが，とにかく，スイッチの状態がすぐにモジュールの実行に反映されます。このようなことを行おうとすると，暗転中はマウスを動かしただけで，モジュールの実行が終了してしまいます。しかし，デモモードでは，モジュールの実行は画面暗前上の「STOP」ボタンを押すことでしか，打ち切られません。ですから，この場合のようにコントロールをいじりながら，モジュールのパラメータを決定することができるようになるわけです。

図1　画面暗前とモジュールの関係

図2　ウィンドウ各部名称

来月号の予告

ここではスクリーンセーバーの簡単な使用方法と，そのモジュールのひとつであるFISH.Xの使い方（と，その注意点）を説明しました。

FISH.Xは，とにかく凄いモジュールを，と横内氏が作ってくれたものです。確かに凄いモジュールですが，モジュールの作成に関して参考になるようなプログラムではありません。私も横内氏にきっぱりと「細かい解析はおそらく無理です。なにか問題があったら連絡してください」といわれてしまいました。

来月は，あまり凄くないけど参考にはなる残りのモジュールについて，スクリーンセーバーの技術的解説と共に説明します。

というところで，また来月。ガシガシ(あわててアクセラレータを作る音)。

SX-WINDOW開発支援ツール

ウィンドウデザイナ(暫定版)

Ishigami Tatsuya 石上 達也

**SX-WINDOWのプログラミング環境はまだまだ整っていません
面倒なことはできるだけ自動化したい
とりあえず使えるウィンドウエディタを作ってみました**

MS-Windowsが3.1にバージョンアップされました。ファイルマネージャが少しは使えるようになったとか，マルチメディアに対応したとか，いろいろ特徴があるのですが，私にとって一番重要だったのは，「マインスイーパー」というゲームが標準でついてきた，ということでした。

大きな声ではいえませんが，私の場合，MS-Windowsをいじっている時間の約8割は，このゲームをやっています（2割はシステムの調整に苦しんでいます)。

しばらくの間，486DX/33＋S3というマシンが，マインスイーパー専用マシンとなってしまいました。

そして，この中毒症状がなんとか治まったところで，SX開発キットの少しバージョンが上がったものが送られてきました。つらつらつら～っと中身を眺めていると，その中に「爆弾ゲーム」の文字が，にっこりと私に微笑みかけてくるではありませんか。

このところ，私は1日に3時間しか寝ていません（わはは)。

さて，そのSX開発キットについてくるリソースエディタが，「あーあ」だった場合，私がウィンドウエディタを作る，と5月号に書きました。完全な「あーあ」状態ではないのですが，どうも，SX-BASICのフロントエンドとして使えるようなものではないようです。

たとえば，ダイアログリソースの編集の場合，ポチポチと，キーボードからテキスト形式で属性や座標を入力し，それを表示させながら，試行錯誤を繰り返すものみたいです。

まだ，完成版が手元にあるわけではないので断言できませんが，あの広告写真にあるグラフィカルなウィンドウは，キーボードから打ち込んだテキスト形式のデータを表示するため「だけ」のものでした。現在のところは，マウスひとつでらくらく操作ってな感じで，ウィンドウをデザインすることはできません。

「プログラムの編集，リソースの作成，コンパイル，デバッグといった一連の作業をSX-WINDOW上で効率よく実行できます」と広告にあるのに，なぜかできません。できませんとも。ええ，できません。できるかなあ。できるといいなあ。なぜできないっ！　開発キットの発売まで，しばらく時間があるそうですから，改善されているかもしれません。期待しましょう。

ウィンドウデザイン

コマンドシェル上では，文字は基本的に左から右へと一方的に流れ，適当なところで改行し，また一方的に左から右へと流れていきます。カーソルの位置を明示的にいじってやらないかぎり，この規則は守られます。

それに対し，SX-WINDOWのウィンドウの中には，あっちこっちに，いろいろな文字が書かれていたり，ボタンがあったりします。文字の大きさも一定ではありませんし，さまざまな種類や大きさのボタンが縦に並んでいたり，横に並んでいたりします。

SX-WINDOWにも，カーソル位置というか，ペン位置のような概念はあるのですが，あまり，これにこだわると面白味のないウィンドウになってしまいます。そんなわけで，文字列を表示したりする場合には，たいてい，一緒にその座標も設定してやります(BASICでいうと，PRINT文よりもSYMBOL文とペアになって出てくる，ということ)。

BASICでプログラミングする場合は，画面のデザインといっても，文字の大きさは同じですし，キャラクタ単位でしか指定しないし，わりと変更が簡単ですし，そういうわけで，実行しながら適当に改造していくと，いつの間にかプログラムはできあがっていました。

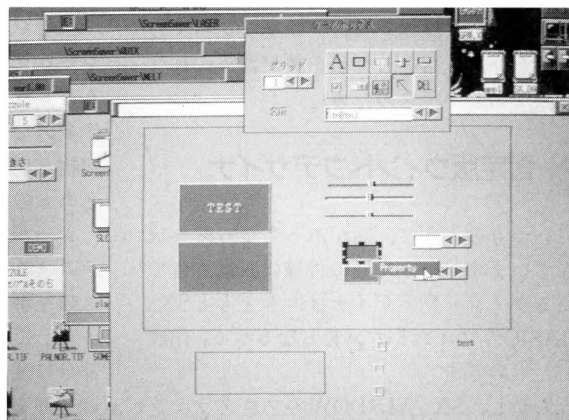
しかし，SX-WINDOWの画面はそんな風にはデザインできません。

まず，扱う情報の量が違います。同じ文字列を表示するにしても，背景色やら描画色やら，フォントの種類やら，と，決めなければいけないことが，どばっと増えます。

そして，SX-WINDOWのはX-BASICみたいなインタプリタ言語がありませんから，適当にデータをいじりながら，適当に実行という技が使えません。

なかには，スタッフの牛嶋氏のように，デバッガを立ち上げて，なにやらブツブツつぶやきながら，直接メモリをエディットして，疑似インタラクティブ環境を強引に構築してしまう人もいますが，普通の人には真似できません。

しかも，SX-WINDOWのアプリケーションで使える唯一の高級言語はC言語ですが，いったんプログラムを変更してから，実行できるようになるまでには，どんなに頑張っても1分はかかります。

こんな環境のなか，それでもSX-WINDOWのアプリケーションを作ろうとする

これが基本画面

人は，いきおい次の2つの方法を選ばなくてはなりませんでした。
1)　それでも，データをいじりながら，プログラムの変更→コンパイル→実行，というサイクルを延々と繰り返す
2)　プログラムに入る前に，方眼紙にあらかじめウィンドウのデザインをしておく。そして，プログラミング時に方眼紙のマス目をもとに，座標データを得る

特に，2)はまだBASICがあまり普及していなかった頃,FORTRANでプログラムを作る際によく行われていた手法です。このような面倒な作業を自動化するためにコンピュータは進化してきたようなものなのに，また10年分近く，その流れを逆行しています。

そんなわけで，SX-WINDOWのプログラムを作るには，まだまだ忍耐力が必要なのです。多機能なサービスコールやライブラリ，スケルトンの使い回しやらによって，かなりの部分で助けられてはいますが，面倒なことには変わりありません。

イベントドリブンの考え方や，ハンドルを通してのメモリの扱いやら，面倒臭い部分はほかにもいっぱいありますが，全体の何割かは画面のデザインです。

じゃあ，どうしたら画面のデザインが楽に行えるようになるかといえば，話は簡単で，それこそWYSIWYG(知らない人はいないと思うけど，What You See Is What You Get＝見たものそのものが手に入る）すればいいのです。

適当にウィンドウをデザインしてみて，
「あ，この文字列はもう少し左だったほうがよかったな」
と思えば，マウスで文字列をひょいっとつまみ，少し左にずらしてやることができれば，かなりウィンドウのデザインは楽になるはずです。

それなのに，シャープも計測技研もINCも，そういうツールを作ってくれないのです。そういう怒りを胸にアクセラレータの記事を今月も落とした私を誰が責められよう（反語です。念のため)。

暫定版ウィンドウデザイナ

いつものように，話があっちこっちへ飛んでいますが，今月号の付録に収録されているウィンドウデザイナは，もともとSX-BASICのフロントエンドとなるべく，作成したものです。

しかし，SX-WINDOWのプログラミング環境がこんなにも過酷なのに，改善される見通しがありません。ないよりはマシだろう，と思って，とりあえず，いま動いている部分に手を加えて，暫定版ウィンドウデザイナとして発表します。

なぜ，これが完成版ではないかというと，最終的にはC言語のソースを吐き出すようなウィンドウデザイナを考えているのです。ボリュームやオルタネートスイッチのプロパティ設定（後述）を行おうとすると，まっさらなウィンドウが開いてしまうのは，そのためです。いずれ機会を見て，このウィンドウには，いろいろとプロパティ設定用のコントロール類をつけようと思っています（初期値，最大値，最小値など)。

宣伝するようで気がひけるのですが,（開発キットと同様，遅々として制作が進まない）SXのMOOKには完全版が掲載できるようにする予定です（予定は未定にして……)。

アイテムの配置，変更，削除

まず，SX-WINDOW上からでも，コマンドライン上からでも，WIND.Xを起動します（その前に付録ディスクを解凍するのですが，その手順は，前のほうの記事を読んでください)。

すると，ウィンドウデザイナと書かれたまっさらなウィンドウと，ツールボックスと書かれた少し小さなウィンドウが表示されているはずです。

大きなほうのウィンドウが，これから作業を行うカンバス（みたいなもの）で，小さなほうのツールボックスウィンドウがパレット（みたいなもの）に相当します。パレットの上から適当な，画材を選び，カンバスの上にパタペタと貼っていきます。

パレットはカンバスのサブウィンドウになっていて，カンバスのウィンドウがアクティブになったときのみ，画面に表示されます。

作業を行うには，まず，カンバスの左端をマウスでつかみ，適当な大きさになるようにドラッグします（この作業は，一連の作業のなかでいつでも行えますが，一応縁起物ということで，最初にこの作業を行うとあとの動作がスムーズに行えます)。

ツールボックスのなかから，適当なアイテム（ここでは，矢印とDEL以外）をマウスの左ボタンで選びます。

そして，そのアイテムをカンバスのどの部分に配置するかを指定します。カンバスの一部をマウスの左ボタンでクリックし座標を決定し，そのままドラッグして大きさを決めてやります。

もし，そのようにして配置したアイテムの大きさや配置が気にいらなかった場合には，ツールボックスから矢印のアイテムを選択し，変更を加えたいアイテムを左クリックします。すると，選ばれたアイテムは，その周囲を8個の小さな正方形とともに黒い枠で囲まれます。これが，アイテムが選択された（アクティブになった）というサインです。

選択したアイテムを移動したい場合には，その枠の中心をマウスの左ボタンでドラッグします。ドラッグしようとすると，ウィンドウ上からアイテムが消え，代わりに消えたアイテムと同じ大きさを持った，灰色の枠がマウスの動きにあわせて移動するので，適当な場所までドラッグしてください。マウスの左ボタンを離せば，灰色の枠は消え，先ほどのアイテムが現れます。

また，アイテムの大きさを変更する場合には，そのアイテムの周囲に表示されている小さな黒い正方形（以後，ハンドルという）をマウスでドラッグしてください。アイテムの移動と同じ要領で，アイテムの大きさが変更されるはずです。

アイテムを削除したい場合には，移動，大きさ変更と同じように，削除したいアイテムを選択し，ツールボックス中のDELボタンを押してください。

ツールボックス

ツールボックス（＝パレット）の道具には次のような種類があります。これは，このウィンドウデザイナで扱えるアイテムの種類です（以下，左上から順に右へ)。

●テキスト

文字列を張り込む枠です。文字列そのものではありません。枠を配置したあとで，プロパティ設定（後述）を行い，枠の中に文字列を注ぎ込んでやる，というスタイルをとります。

枠の中に，文字列は1行だけ，中寄せで，入れてやることができます。

●レクタングル

長方形の枠を，描画することができるようになります。複数のスイッチをひとまとめにしたり，ウィンドウのデザインにメリハリをつけるときに使用するとよいと思います。

また，このウィンドウデザイナでサポートされていない機能（ピクチャーボタン）を使おうとする場合に，とりあえず座標を目立たせておくのに便利です。

●**標準ボタン**

標準ボタンを描画できるようになります。

●**ボリューム**

ボリュームボタン（スライダーボタン）を描画できるようになります。

●**オルタネートボタン**

オルタネートボタンを描画できるようになります。

（下段左にいって）

●**チェックボタン**

チェックボタンを描画できるようになります。

●**アップダウンボタン（数値調整ボタン）**

アップダウンボタンを描画できるようになります。

●**ビットマップ**

絵を表示できるようになります。ここで表示できる絵とは，SX-WINDOW上のパターンエディタ，またはEasypaint SX-68Kで作成したPT4形式のものです。

画面に，すぐには表示されませんが，後述するプロパティ設定を行うことによって，表示されます。

●**選択**

一度配置してしまったアイテムを移動したり，サイズを変更したり，プロパティを設定/変更するときに使います。このツールを選択している状態で，マウスカーソルを特定のアイテムの上に載せて，左ボタンを押すと，そのアイテムが選択されます。

●**削除**

特定のアイテムが選択されているときに，このボタンを押すと，そのアイテムを削除することができます。

図2　グリッド

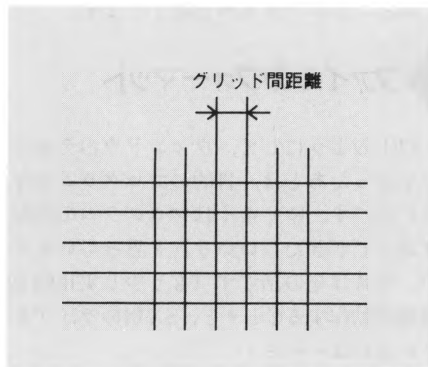



図3　整然とボタンが並んでいる場合

●**グリッド**

グリッド（Grid）とは，日本語で「格子」のことです。わかりやすく日本語で格子と書くと，ますますわけがわからなくなってしまうので，カタカナでグリッドと書きました。

このグリッドと書かれたアップダウンボタンで，ウィンドウデザイナのグリッド間距離（単位はドット）を設定します。グリッド間距離とは図2に示すような2つの直線の距離です。この設定は，キーボードのSHIFTキーを押しているあいだだけ有効です。

たとえば，図3のように同じ大きさの標準ボタンを3つ縦に揃えて並べるときに使います。普通にマウスを操作すると，このように配置することは困難です。不可能ではありませんが，CRT上の1ドットの差を読み取れる眼力が必要となってきます。1ドットの違いがわからない人でも10ドットの差ならなんとかわかるでしょう。で，このグリッド指定を10に設定します。

そして，通常と同じようにアイテムを配置します。ただし，マウスをいじるときには，キーボードのSHIFTキーを押します。このようにすると，ウィンドウデザイナはマウス座標を10の整数倍としてしか認識しないようになります。つまり，アイテムは，(10×N, 10×M）のような座標にしか配置できなくなるわけです。

こうすれば，図3のようにアイテムを配置するのもそう困難なことではありません。

●**アイテム選択ボタン（アップダウン）**

通常，アイテムの選択は選択ボタンを押して，マウスの左ボタンを押すことによって行います。ところが，このような方法では選択できないアイテムが存在する場合があります。

実は，このウィンドウデザイナでは，内部処理的にアイテムが「奥行き」を持っています。奥行きといっても，3次元的なものというよりは，SX-WINDOWでいうところの「ウィンドウ優先順位」くらいに考えてください。SX-WINDOWでは，大きなウィンドウの下に隠れてしまった小さなウィンドウを直接マウスでいじることはできませんでした。手前のウィンドウのほうが優先順位が高いからです。このようなとき

図1　ツールボックス

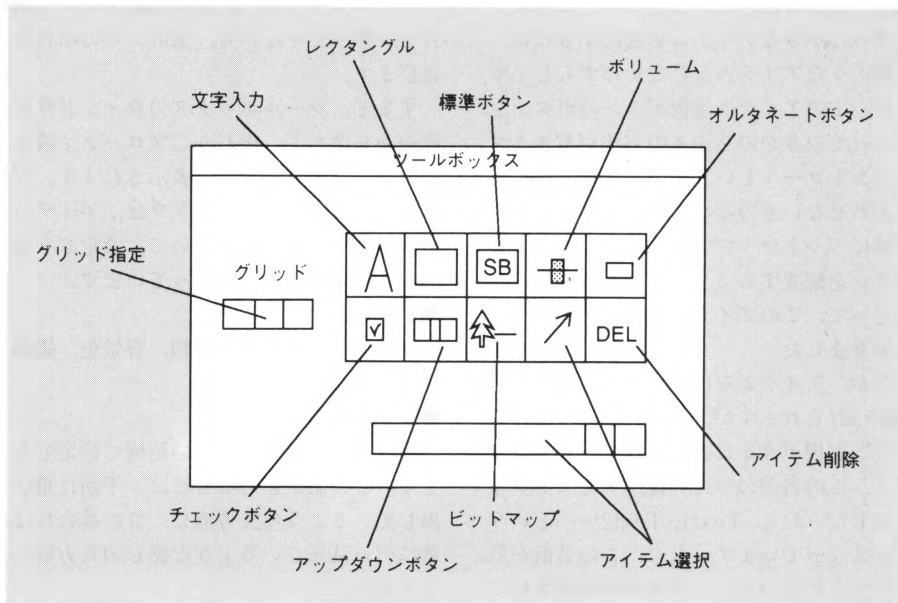
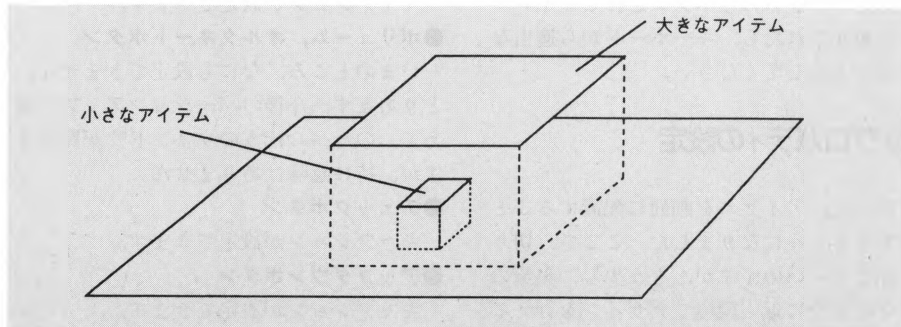
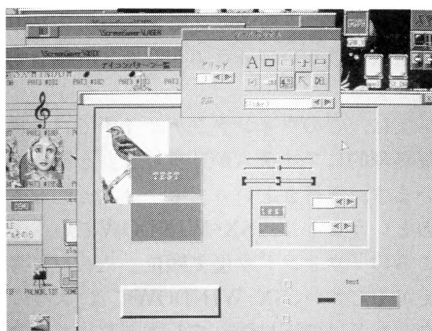



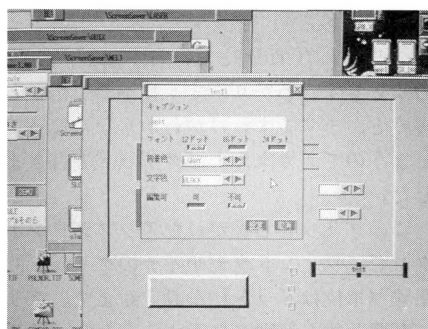
図4　マウスで選択できないアイテム

ボタンにキャプションを入れる

ビットマップを入れる

には，手前のウィンドウをいったんどこかにずらしてやるか，ページアイコンをいじってやればいいのです。これと同じように，図4のように大きなアイテムの下に隠れてしまった小さなアイテムを選択するには，一度大きなアイテムをどこかにずらしてやるか，このアイテム選択ボタンのボタンをいじってやるかの2通りの方法があります。

コントロールというものは，基本的に重ねあわせないものなのですが，ウィンドウ全体にビットマップを表示して，その上にボタンを配置するような場合もあるだろうと思って，このアイテム選択ボタンをつけておきました。

なお，アイテムを作成すると自動的に名前がつけられますが，その名前は，このアイテム選択ボタンに表示されているとおりです。この名前のつけ方は，テキストフィールドだったら，Text1，Text2……というようになっています。もし，この名前が気にいらない場合は，この名前が表示されている部分が編集可能な文字列となっていますので，マウスの左ボタンを押しキャレットが表示されたら，キーボードから適当な名前を入力してください。

プロパティの設定

以上で，アイテムを画面に配置することができるようになりました。ここで，切り上げてもいいのですが，もう少し，実際のウィンドウに近い環境でデザインを行える

文字のプロパティ

ように，アイテムのプロパティを設定できるようになっています。

たとえば，文字列の色を変更してみたり，標準ボタンの中に「確認」だとか「取消」のような文字を入れてみたりすることができます。

このように，アイテムに付属する文字列（以下キャプション）や色などのことをプロパティと呼びます。設定できるプロパティの種類はアイテムによります。

プロパティを設定するには，まず，プロパティを設定したいアイテムを選択します。そのアイテムがアクティブになったら，マウスの右ボタンでメニューを表示します。このメニューには，まだPropatiesという項目しかありませんから，素直にその項目を選びます。

すると，ツールボックスのウィンドウが画面から消えて，代わりにプロパティ設定用のウィンドウが画面に表示されます。プロパティ設定用のウィンドウは，アイテムの種類ごとに用意されていて，設定できる内容は以下のとおりになっています。

●テキスト

文字列，フォントの種類，背景色，描画色

●レクタングル

「彫り」の深さ。－5～5の範囲で設定できます。この値が正の場合には，手前に飛び出しているような長方形に，負の場合には，奥に引っ込んでいるような感じの長方形になります。

●標準ボタン

キャプションが設定できます。

●ボリューム，オルタネートボタン

いまのところ，なにも設定できません。とりあえず，今後のバージョンアップに備えて，のっぺらぼうのウィンドウが開きますが，特に意味はありません。

●チェックボタン

キャプションが設定できます。

●アップダウンボタン

キャプションが設定できます。

●ビットマップ

指定された座標に表示するPAT4形式の絵のデータが入ったファイル名を指定することができます。ファイル名を指定すると，アイテムの大きさは，強制的に絵の大きさにあわせられます。

ファイルの保存

なにもアイテムを選択していない状態で，カンバスのタイトル部分へマウスを持っていくと，右ボタンで，ファイル関係のメニューが開きます。

メニューの項目を上から説明すると，

●ロード

つぎに述べる「セーブ」でファイルに保存したデータを読み込みます。それ以外のファイルを指定しないでください。あまりエラーチェックが厳しくないので，下手をするとシステムエラーが発生してしまいます。

この項目を選択すると，ファイルネームを聞いてくるウィンドウが開きますので，そこにロードしたいファイルの名前を入力してください。

●セーブ

現在のアイテム，そのプロパティの内容，ウィンドウの大きさなどの情報をファイルに出力します。ファイルのフォーマットは後述します。

この項目を選択すると，ファイルネームを聞いてくるウィンドウが開きますので，そこにセーブしたいファイルの名前を入力してください。

●終了

ウィンドウデザイナを終了します。

ファイルのフォーマット

以上のようにして，ウィンドウのデザインを行ったあとは，目的のプログラムを作るだけです。ゆくゆくはプログラムの自動作成までできたらいいな，と思っていますが，それは先の話です（もう少し実用的な環境で動作するC＋＋と，SX用のライブラリがあれば……）。

今回のバージョンでは，プログラミングは人間が行わなくてはいけません。

エディタの上のほうにちょこっと，ウィンドウデザイナの出力したファイルを表示させておき，それを横目で眺めながら，プログラミングを行っていく，というスタイルになると思います。

そのようなことを行うには，ウィンドウ

デザイナが出力するファイルの読み方を知らなくてはなりません。
ウィンドウデザイナが出力するファイルのフォーマットはテキスト形式で，以下のようになっています。

●ID, 名前, (X1, Y1, X2, Y2), パラメータ

IDは1～8の範囲の数値で，割り当ては以下のとおりになっています。

1：テキストフィールド
2：レクタングルフィールド
3：標準ボタン
4：ボリューム
5：オルタネートボタン
6：チェックボタン
7：アップダウンボタン
8：ビットマップ

名前は，アイテム選択ボタンに表示されていたものと同じです。
続く，(X1,Y1,X2,Y2) は，アイテムの配置されたレクタングルの座標です。
パラメータは，アイテムの種類によって以下のようになっています。

●テキストフィールド

n1,n2,n3,文字列
n1：背景色（0＝白， 1＝薄いグレー，2＝濃いグレー， 3＝黒）
n2：文字色（同上）
n3：フォントの種類（0＝12ドット，1＝16ドット，2＝24ドット）
文字列：キャプションに指定したもの

●レクタングルフィールド

n：「彫り」の高さ。負の値で彫り込み，正の値で飛び出してくる。範囲は－5～5。

●標準ボタン，チェックボタン，アップダウンボタン

文字列：キャプションに指定したもの

●ボリューム，オルタネートボタン

パラメータなし。

●ビットマップ

文字列：指定したファイル名

プログラムについて

このプログラムは，評価用のSX開発キットのなかに入っていた「簡易ドロー」を改造して作成したものです。変数名や関数名に，一部ちぐはぐなところがあるのはそのためです。まだ開発キット自体が発表されていないのですが，待ち切れませんのでとりあえず先に発表します。
また，「簡易ドロー」はOBJR型のプログラムで，グローバル変数を一括して，構造体で持っていましたが，私は追加した分のグローバル変数は，そのようになっていませんので，ウィンドウデザイナはOBJC型のプログラムです。
ウィンドウデザイナ自体が，まだ暫定版ということですし，製品版のSX開発キットには，簡易ドローの説明がどのようになるかわからないので，具体的な解説は，完成版ウィンドウデザイナができてから行うことにします。

最後に

このプログラムは，あわよくばSX-BASICのフロントエンドとして，あるいはQuick-C For Windows（以下QC/W）についてきたようなCase:Wのようなものを目指して作成してきた途中のものです。途中のものですから，不完全な部分が多々見受けられます。
しかし，機能が不完全でも，ウィンドウデザイナがあれば，プログラムの開発が大幅に楽になるだろうということで，今回の付録ディスクに収録させていただきました。
最近，私はMS-Windowsを意識して，プログラムを作成しています。MS-Windowsは合計で100万本も売れ，ウィンドウシステムのデファクトスタンダードとなりつつあります。
しかし，個人的使用に限っていえば，Visual BASIC，QC/W，Word，Excelみたいなものがあれば，ウィンドウシステムは，なんだっていいのです。
QC/Wというのは，Cコンパイラ＋パターンエディタ＋デバッガ＋Case:Wです。前から2つ目までは，すでにSX-WINDOW用にもありますし，SX開発キットが発売されれば，デバッガはQC/Wにも劣らぬものが手に入るわけです。で，もちっと私が頑張れば，QC/Wくらいの環境がSX-WINDOWにも揃うことになります。
ついでに，もう少しホラを吹いておくと，MS-Windowsの主なソフトは，
Word→シャーペン
Visual BASIC→(大幅に機能は落ちるけど，Mookに掲載予定の)SX-BASIC＋(そのサポートツール群)
となります。
DTPR用のソフトやレタッチ系のソフトは個人ではしばらく用がないとして，残りはExcelです。荻窪師匠もいってましたが，SX-WINDOW用の場合，コテコテの超多機能ソフトである必要はありません。ピリリと小粒な機能がひととおり揃っていればいいのです。
誰かSX用の表計算ソフトを作ってみませんか？

＊　＊　＊

ウィンドウデザイナの使用決定に際して，INTERVIEWの使い方を教えてくださった牛嶋健雄氏，Hyper Cardのマニュアル＆参考書をどばっと貸してくださった元編集のA氏に，謝意を表します。

謎のONKAI.Xについて

楽譜エディタがほしい，と，他人にもちかけてもどうも逃げられる。なら自分で作るか……とも思ったが要求する操作性を満足させるにはちょっと道が遠い。で，SX-WINDOWだ。見れば，エディタ部で必要なことのほとんどがすでに実現されて転がっているではないか。
で，仕様を削れるだけ削る。犠牲者（石上君ともいう）に話をもちかける。
「え？ そんなんでいいんですかぁ」
ということで，SX-WINDOW史上もっとも簡単なプログラム（かな？）が誕生したわけだ。
このプログラムはウィンドウを持たないSX-WINDOWアプリケーションである。どんな機能を持っているかというと，「画面上のY座標に応じた音階を発生する」ことができる。
で，どこがどの音程になっているかというと，それ用の背景が用意されているのでそれを目安に設定することができる。要するに背景に楽譜を描いて，シンボルで音符を置く。そのシンボルに設定されるべき実行ファイルがこの音階発生プログラムとなる。背景に置いたシンボルならばマウスひとつでエディットも思いのままだ（一応は……）。
使い方を解説しよう。まずコトロールパネルを操作し，背景設定を実行。付属のSCORE.PT4を張り付ける。
次にシンボルトレイに新規シンボルを設定する。付属の音符データを使うことが前提となっているが，種類が足りなければ適当に拡張することが望ましい。シンボルはたくさん登録しておこう。名前はなしで，実行ファイルはONKAI.Xである。
あとはこれを画面の好きなところへ並べておけばよい。一応ダブルクリックで音が鳴るし，自動的にセーブだってされる。楽譜の印刷もできる。インチキ臭いが，これでも実用品だ。ちょこっと音取りしたいときには使えるかな？
楽譜エディタというと嘘っぽいが，X68000でこれより綺麗な楽譜を描くツールはいまのところない。最大の欠点は使用する音符の数だけシンボルを定義しなければならないことか。

＊　＊　＊

印刷まで考えると現在の設定でいいのだが，実用上は音符シンボルの色をダークグレイに変えたほうがいいだろう（反転すると見えなくなる)。パターンを変更して，ラインペイントで白の縁取りをするなどすれば反転しても大丈夫なのだが，五線に切れ目が入ってしまうので美しくはない。ああ，早くちゃんとした楽譜エディタがほしい。　（中野修一）

ダイブ野郎女郎に贈る悩殺環境

FISH.X

Yokouchi Takeshi 横内 威至

SX-WINDOWらしかぬ反則スレスレの力技の数々
そして画面中をのんびり泳ぐさまざまな魚たち
目が離せなくなって作業が進まなくなったりして……

すでにエラ呼吸もマスターした貴方のために，もしかしたら陸でも心休まるひとときが得られるかもしれないデモを作った。

ことの始まりはたしか今年の1月ぐらいに中野氏と石上氏がなにやら怪しい話をしていたところからであった。

「こう，熱帯魚がひらひらと泳いでるとステキじゃないですか」

「あ，ちょうど無職の横内君がいるじゃないですか。たまには社会的なことをさせましょうよ」

だが，当時どうせすぐ飽きるのにダイブなんていいかな，と思っていた俺は熱帯魚というのを熱帯海水魚にして欺いてしまったのである。

ディスクの締め切りは当時5月前半といわれていた。それに間に合わせるかたちで，5月の連休に入った頃にやっと本腰を入れて開発を始める。初めて図書館という，俺には一生縁のなさそうな，そのあまりに崇高な空気を以て俺を死に至らしめる館に入館して資料を持ち出し，結局あまり使わなかったスキャナを編集部からお借りして，いよいよ本気。

このデータのせいだとか，その他の事情により，ディスクの締め切りを大幅に破ったことをここにてお詫びしたい。付録ディスクが遅くなったのは俺のせいかもしれないとい ま本気で反省している限りである。

内容なんかのこと

えー，ウィンドウでトースターが飛ぶなんてゴージャスなものではなくてなるべくリアルに描いた魚が呑気に泳ぐだけ。

本当はコマンド一発起動のデモで，店頭デモ用にロゴの入った奴なんかも最初はあった。X68000ならではの超スプライトで画面狭しと暴れ回る海中生物たち。X68000以外では決して許されない，他を圧倒してビビらせてX68000の魅力でメロメロにして買い替え促進させるようなデモ，これを念頭において作った。

でもまだまだだな。まあとりあえずはこのくらいで一段落とさせてもらいたい。でかい魚さえいなけりゃこの2倍の量で量産状態であったのだが。やっぱりスプライト2倍はほしいね。

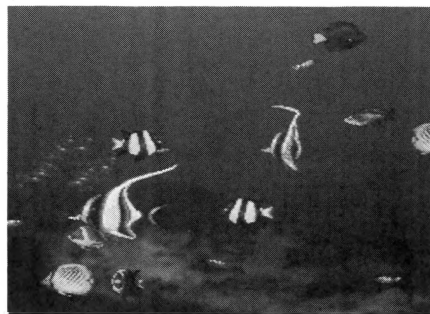
おっと，X68030でももちろん動く。困ったことにキャッシュが悪さするから内部で自動判定してキャッシュを殺している。以前あったX68030自動判定のプログラムを参考にした。キャッシュについては福嶋君，ありがとう。またザンギで対戦しような。

ところでバグはないとは思う。これのおかげでディスクがイカレた，猫がモニタを破壊した，火山が噴火した，夏が寒い，少しは横内を見直した，などの苦情や責任には一切関与しない。以上。

ま，そんなことはいいか。せっかくだからもっとプログラム寄りのことを話そう。

今回は石上氏と俺との合作である。俺はコマンドライン上で動くデモソフトを作り，俺では10年かかっても不可能であろうSX-WINDOWへのコンバートを石上氏にしていただいた。極悪なまでに行儀の悪いカオスソースを，本当によく移植してもらい感謝している。こんなものをスクリーンセーバーのモジュールにしてしまってよいのかどうか不安であった。タスクも奪い，ラスターも奪い，スプライトまで転がしているのにねぇ。

プログラムと姑息な技

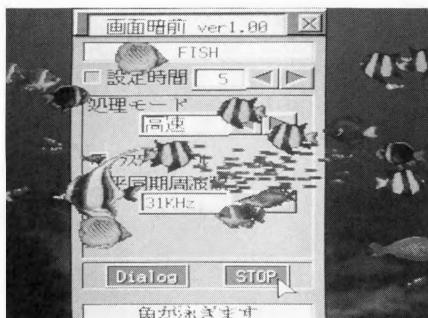
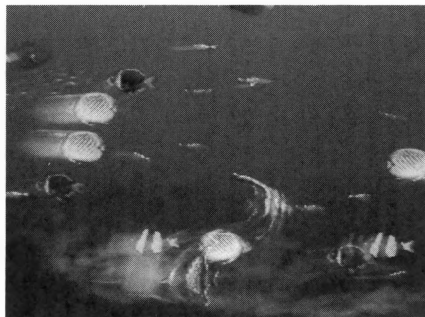
構造は1992年11月号に掲載されたスプライト関係の俺の記事を読めばだいたいわかると思う。ただアレに優先度をつけたり，その他いろいろなタイプで分けてそれなりに管理してやっている。このあたりのことはまああまり関係ない。それでも参考にしたいならばポインタだらけであることに注意して解析なんかもやってもらいたい。ただ，本当に突き詰めたナイスプログラムではないから参考程度にすべし。

優雅に泳ぐ魚たち

となると，困るのは回転ルーチンですかね。一部の方がもしかしたら興味をそそられるルーチンを解説したいと思う。

まだ結構質問がきます技

でも，まず一発目はすごく基本でありながらまだ浸透してはいないラスター割り込みから。

これはCRTCの動作と深く結びついていることは以前の質問箱で解説してあるし有名。あれでわからなければおそらく割り込みという動作自体が理解できてないのだと思う。初心者の方にはキツイかもしれないが，ラスターに限らずもっと一般的な割り込みについて学ぶべきであろう。まあ，要するに普通の動作中にMFPって奴が誘ってくるだけのことだ。

俺が道を歩いているとしよう。俺がCPU（関係ないけどインテルではない）で道がプログラムだ。しかし俺はとっても美しくカッコイイからいろんな人が声をかけてくる。肥をかけられると最悪。まずこれまた美しい淑女が声をかけてくる。もちろんシカトするはずがない。俺はいったん止まってしっかりと応対する。でもただのタチンボだったから適当にあしらってさらに歩き

デモモードはこんな感じ

当然，動きは滑らか

続ける。次に声をかけてきたのはポン引きだ。ポン引きがつく店はボッタクリという定義に従って無視。あらかじめ俺の耳までは入れぬよういつも心がけている。割り込みをマスクするって奴である。ズバズバ進んでいく。定期的に信号があって足を休めつつラッキーを吸う。都会の臭い大気と入れ換えにヤニを吸収してリフレッシュする。タイマ割り込みのようなものだ。続いて声をかけてきたのはパンチパーマがワイルドなヤクザ。無視しようにも強制的に止められる。下手すればそのまま死亡，リセットにて終了であった。

どうだ，これでもうバッチリ割り込みマスターだ。X68000において，この数々の刺客を送り込んでくるのがだいたいMFP。キーボードだとかタイマだとかラスター，その他いろいろ。それぞれに対応した割り込みルーチンを別に用意してやればよい。いらないのはマスクする。リセットなんかは強制であってマスク不可能。細かい設定が多すぎるから『Inside X68000』なんかを参照すること。これで決まりだね。

さて，ラスター割り込みはただ画面がゆらゆらするだけの古代から伝わる処理以外にとても素晴らしい技がある。ラスターによって画面モード，優先順位，さらにいじるとパレットなんかも変えてしまえるのである。

実例をひとつ挙げると，ストライダー飛竜が挙げられる。見せ場である2面の大滑降，木がグラフィックだけど自分より手前，つまり特殊プライオリティを使用しているところである。これは実に扱いにくいもので，たとえばスコアなんかをテキストに書いてもとにかく指定したグラフィックが優先されてしまう。スコアあたりの部分はグラフィックオフにし，その下あたりでラスター割り込みしてオンにするのである。美しい技である。

そしてもうひとつ，スピード狂のプログラムでは守らねばならない鉄則がある。スプライト関係なのだが，実はスプライトのアクセスは遅い。これはスクロールレジスタ，そして当然PCGエリアも含まれる。なぜならこれは常にCRTCがいやらしくちょっかいを出しているからである。垂直帰線期間とて油断ならないらしい。だからせめてこの無意味な期間にはCRTCのアクセスを殺さねばならない。これは以前にも説明したが，前回ではサンプルがなかった。でも今度はスペックが苦しいぜ，ということで当然使っている。リストのMONITOR ONあたりをよく見てもらえば理解できるであろう。大流行の格ゲーを作りたいなんて貴方ならもう基本だね。

ラスタースクロールに限定した話をちょっとすると，これでY方向のスクロールを変えることができるのはスプライト，BGのみである。グラフィックとテキストはY方向固定。垂直帰線期間の設定が有効とされる。

それと注意すべきは処理落ちの動作である。割り込み処理ルーチンで，ただ単純に次は4ラスター下だぜ，と設定するといつか画面遙か下方で割り込め，となる。でももちろんラスターはそんなにないから無視される。処理落ちしててもCRTCはさっさと次の画面をスキャンするが，ここでスクロールレジスタの値は最後に設定した値から更新されないのだ。当然画面は時分割でブレを起こす。こればかりは皆の好きなように処理してほしい。設定ラスターをandで戻すだとかでうまく，それでいて高速に処理しなくてはならない。

コナミもやらなかった変態技

スプライトPCGアドレスは変態的な構造をしている。水平型でも垂直型でもない困った構造だ。だから単純な縮小回転は不可能であると思われた。でもやっぱりやる。

まず普通に考えよう。元絵をどうやって回転させるか。これは一般的には逆変換といわれている方法をとる。これは回転先が完全な矩形領域，というか画面のXY軸に平行な辺を持つ矩形領域にする。このほうがコンピュータはやりやすい。

簡単に考えれば表示を斜めラインの集まりではなくただのXYループで表示できるからだ。この方法のマズい点は回転元のデータが回転先の対角線の長さを直径とした円になることである。当然元絵自体をこれにあわせなければならない。

さて，これで考えると元絵からデータをうまくサンプルしてきて順番に回転先に置いていくだけ，ということになる。では次の問題はどうやってサンプルしてくるかだ。回転先の左上にあたる点をまず必要とする。一次変換で一発である。あとはそこを起点としてベクトルを足してやれば順番どおりにサンプルできる。回転先1ライン下は時計回りに90度のベクトルを加えるだけ。ベクトルなんてシブイぜ，と敬遠するな。パラメータX，Yにただ三角関数を足していってやれば完全なベクトル演算である。

小数点が出るなんて泣けるぜ，なんていうなよ。68000は小数演算を基本演算としてサポートしていることを知らないのか。68000でブレゼンハムなんて似合わないぜ。SWAP命令4クロックで固定16ビット小数点がモロだ。ロングワードで下位ワードを小数，上位を整数にすればきわめて効率よさそうに見えるだろう。

まあこの場合，本当はレジスタが余っていればブレゼンハムによる検出も悪くない。クロックを数えてないからどちらが速いかわからないが条件分けが比較的面倒。だが落とし穴は縮小ルーチンである。縮小すると次にサンプルすべき点は最悪何十ドットも飛ぶ。ブレゼンハムでは移動が1単位以外は許されない。ましてこれを整数，小数に分けたりすれば単純計算で2倍の重さ。おそらくレジスタ数もきつくなる。やっぱり固定小数点のほうがエレガントに俺には思える。ということで小数点でベクトルが楽，ということにしておく。

寄り道してしまったが，このベクトルの長さを変えれば拡大縮小ができることは明らかであろう。拡大はベクトルを1ドット分以下にすればよい。縮小は逆。内部では当然$10000を1ドットに相当する大きさとしてある。さて，ここで恐ろしいのは縮小。ベクトルを大きくする，ってことはサンプルすべき領域をはるかにオーバーすることになる。浅はかに考えると，これはもっとも縮小したときにサンプルする領域をすべてクリアしておかねばならない。最高で64分の1までなら元絵の64倍の大きさの領域を確保しなくてはならない。

というということでデータを作るときにマスクをしてやる。あらかじめ縮小率がわかっているからそれに見合った円でマスクをすればよい。これはリスト（SP2ROT.H）の中に入っているから興味がある人は好きなように。どうせ64段階だから自力で作る。

ここまでで回転縮小の基本はわかったであろう。ちなみに拡大をサポートしていないのは明らかである。回転先がスプライト4×4の領域と固定だから拡大が意味をなさないのだ。

さて，ここまではごく当たり前のことを解説した。問題はスプライトはグラフィックのようにアドレスが整然としていないことである。これゆえに普通はスプライト回転なんか無視されるのである。どうやったかというと，テキスト画面をワークとして使うことでうまく処理したのだ。

テキストは都合よいことに1バイトで2ドット分であり，しかも1バイト制御で上位4ビット，下位4ビットに分けてアクセスできるようマスク機能が用意されている。そう，つまりサンプルしてきたドットを上位4ビット，下位4ビットと交互にワークに書いていくのだ。サンプルすべきドットは1バイトで16色，4ビットにしてあるが，これを上位4ビットにも同じデータを書いといてやれば毎回ローテートさせる必要も失せる。

ここまでではまだスプライトデータにはならない。だがすでに1バイトで2ドットという，もっとも重い部分の処理はできあがった。これをスプライトに登録すればよいのだ。あとはスプライトのアドレスに対応させた順番に転送させるだけである。以上がスプライトのための拡張である。当然，まずワークにデータを作り，そしてさらにそれをPCGエリアにうまく転送するわけだからグラフィックの回転に比べてかなり重い。

内部では姑息なことに，処理を何回かに分けている。64ドットのうち32ドットまでを回転させたらいったんリターンし，次で残りを回転させて登録，そしてその次でやっとスプライトに登録，って感じ。中でそうやってるかは忘れたが，これでほかを1/60秒で処理しつつ回転の奴も加えることができている。

その他基本的な高速化は当然必要。まず回転先の横1ライン分のアドレス移動テーブルを最初に作る。当然ループ展開をしておく。先ほどのマスクも実は半分ウソ。マスクはしない。あらかじめ何ドット目までは表示なし，そこから何ドットデータをサンプルしてその後何ドットまた表示なし，なのかを調べておく。そうしておいてそれぞれ別々のループ展開を作っておく。これはすべて64ドット分用意してあり，表示に該当するようにうまくRTSを書いたり消したりするだけである。以上をYでループさせれば終わり。

このルーチンでさえあまり突き詰めてはいない。まだ目に見える部分，アルゴリズム以前にコーディング自体で微々たるクロックアップは可能。ちょっとだけだが。あとは座標からアドレスを得る計算なんかも参考になるかも。玄人はこんな方法以外も知っているかもしれない。

グラフィックのYの1ドットが一般的なモードで$400であることは基本。これは1024倍すればよいのだが，シフトでは10回分。レジスタ指定のLSL.L Dn,Dnを使えば確かにそうなる。ただしYを入れる前にMOVEQで目的のレジスタをクリアするかEXT.Lで拡張する必要がある。また別のレジスタに10を入れねばならないことにより，いろいろな無駄が生じる。ということで俺が使っているやり方のほうがマシ。SWAPにより16ビット分ローテート。CLR.Wで16ビット分シフトに相当。あとは6ビット下位にシフトすれば同じ結果でより速いはず。このあとXを足すのだが，これはワードでの加算で十分。なぜならX方向の加算でロングワードの境界を跨ぐようなことはありえないのだ。ほかにも高速演算法を知っている人は，情けないけど俺にも教えて。

終わりに

まだ俺のやっていることで参考になるような人はぜひプログラムをいじってほしい。これはゲームのプログラムとまったく同じレベルのものだからだ。なぜか気合の入ったゲーム作成野郎養成講座がないよな。X68000ならマジで熱い分野なのにね。まあ，チャンスがあれば俺もこんなかたちで，一流ではないけど技術を伝えてはいきたいと思う。なんせ俺はプロではないし，実際ゲームがどうやって動いているか習ったこともない。すべて我流だから自分では気づきもしない落とし穴にはまっているのかもしれない。まあ，動けば問題なし。

ああ，そういえばデモの後書きのはずだったよな。デモのことも最後にまとめよう。

ということで取り込みフェチの俺がドットを貪りながら描いた八景島らしき海。まだ取り込みのレベルには達していない。どうあがいても俺はこのレベルか？　やはり

中心にいるのが回転魚

これがエイ

俺の世界を完成させるには完全取り込みが必要か？　現代さらに加速するハイパーリアリズムに俺も乗らねばならない。この程度ではまだ納得できない。あ，でも改めて見てみると俺も感動してしまう。幸い動きがうまくいったようだ。中野氏による再三の変更要請，そして最後に残された「やっぱりエイがなければ話になりません」の呪文。根性入れて20パターンほど最後に描き上げてだいたい完成。中野氏のバックアップのおかげで俺もそこそこ満足できるレベルまで作れたと思う。この場を借りて感謝の念を刻みこみたい。

あ，その他いろいろ意見が飛んでいた。西川氏は「サルがなければハナクソだ」，俺も同感だが金子氏は「ロータスが潜水してなきゃ情報部員失格」，「隠れキャラのトップレスで優雅に泳ぐ俺とブラザーが必要条件」など。駄目。おっと，唯一まともな案は八重垣氏の「常駐してるものによって出てくる魚を決める」ってのがあった。いいかもしれないけど大変。失礼。

やっぱり次やるときは取り込みだな。一度は俺も試したいぜ。やっぱりリアルじゃなきゃいけないよ。教育上でも悪いぜ。ゲーセンにたむろするガキなんかストIIが当たり前だもん。人殴っても痛くもないし死にもしないとしか思ってないよ。絶対。リアルな映像でリアルなスリルを味わったほうがいいぜ。こう，小学校の道徳の教科書にダーティハリーを載せる，なんてのはどうだい？　あ，駄目か。そんななったらもう救いよう，なしだな。

ポリゴン3Dへの期待

SLASHに寄せて

Nakano Shuichi 中野 修一

期待のポリゴン3Dグラフィックライブラリ
それはどのようにして生まれたのか
これから目指すべきものはなにかを探ってみましょう

待望されていた3Dポリゴナイザの登場である。

とりあえず，最初のSIONをお持ちの方は立ち上げてみていただきたい。続いてSION IIがあれば，それも立ち上げてみていただきたい。SION IVで使われているSLASHというのがいかに高性能なものかおわかりいただけただろうと思う。

最近は巷がどうもポリゴンづいている。

CD-ROMと業界一速いというVRAMに，2つの68000CPUとZ80をフルに使ったシルフィード。DSPを積んで初めて家庭用ゲーム機で本格的なポリゴンシューティングを実現したスターフォックス。ちょっと古いが，32ビットのポリゴンシステムでポリゴンパワー爆発のバーチャレーシング。かなり古いが，32ビットプロセッサと演算用に2つのDSP，強力なポリゴナイザを備えたスターブレード。

どうやら3Dポリゴンはハードウェアによる力技のおかげでようやく実現されてきたようだ。

＊　＊　＊

「X68000に3Dポリゴンゲームを」というのは多くのユーザーの悲願だった。

ところが，発売されるフライトシミュレータなどを見ると，重い重い。でもってX68000は遅いといわれるのはどうも納得できない。まあ，はっきりいって，小容量のVRAM，馬鹿でも使える高速なCPU，ハードウェアでの描画機構を備えたPC-9801のほうが遙かに有利ではあるのだが，マシンの性能を云々する前に基本的なことをやっていないとしか思えない部分がある。

実際，PC-9801などで発売される国産フライトシミュレータなどを見ても，不思議なほど重い。多少条件は違うが8086を積んだDynaBookで軽々と動くポリゴンを見ると，日米でソフトハウスの技術力が根本的に違うというのを実感せざるをえなかった。

3Dモノに関するノウハウの蓄積過程で，根本的な部分を持たぬまま作られた製品，それが大半の国産3Dモノに対する感想であった。

X68000で3DといえばMAGICである。

最初のMAGICでは8ビット時代のアルゴリズムを見直しもせず，とりあえずコードを置き換えるだけ，表示に至ってはIOCSコールをそのまま呼び出すという結構人をナメたシステムが採用された。それでもSIONは十分面白いゲームに仕上がっていたと思う。そしてMAGICのコードは最適化されてSION IIができた。しかし，3Dのアルゴリズムに関してはまったく変更は加えられていなかったのだ。

ことの始まり

「そろそろちゃんとしたレースゲームがほしい」

半年ほど前からみんなの心にわだかまっていたのがこれだった。

ラスタースクロールによって道を曲げる。そこに車を乗せる。「車を左右に動かして障害物を避けながらできるだけ車をコース内に入れてください」というゲームではなく，コーナリングが快感であるようなゲームを求めていたのだ。

一般にラスターモノでは，車の挙動が不自然になることは避けられないとされている。しかし，それもやりよう。内部処理次第で，グリップ感やアクセルワークが反映されるようなコーナリングフィールをかなりのところまで再現できそうに思われた。

ということで，悪魔の囁きをもって，違いのわかるハードコーダー横内君にラスター技でバンクを表現する方法，直角コーナーを表現する方法などを吹き込み，「ラスターでの限界を究めようプロジェクト」が密かに開始された。……のだが，頓挫するのも早かった。

路面の表示自体はなんとかなっても，そこに乗せる車の表示がまともにできないということがネックとなる。車のナチュラルな挙動は最低条件であったのだ。

どうせならちゃんとしたシミュレータをということで，いきなりポリゴンシステムの開発が始められることになった。

AMIGAのこと

話は飛ぶが，ポリゴンといえばAMIGAである。

AMIGAにはたくさんの3Dゲームがある。しかし，はっきりいっておくと，見るべきフライトシミュレータというのは2，3本しかない。見るべきレースゲームも2，3本だけだ。

ここでAMIGAのBLITTERという機構について解説しておこう。これはAGNUSあるいはALICEというチップに含まれる，ひと言でいえば「ビット単位のDMA」である。

実はAMIGAには特定のVRAMがない。AGNUSというDMAの化け物にコマンドを与えればメモリ上（Chip RAM）の任意の部分が画面に転送されるという具合になっている。この処理の中核を成すのがBLITTERである。AMIGAの画面表示のすべてを司る部分である。

AMIGAがメジャーなゲーム機となって5年以上，しかし国内のゲーム専門誌などを見るといまだにAMIGAには3D処理用のカスタムチップが入っていると信じている人がいるらしい（あるいはアニメーション専用ICなど）。しかし，表示に関する限り，この「ビット単位のDMA」以外に特殊な機構というのはほとんど見当たらない。

AMIGAの開発目標にフライトシミュレータがビシバシ動くコンピュータというものがあったというのは有名な話だが，その結論としてBLITTERを搭載したというのは必要にして十分，まさに天才的な選択だ

ったといえるだろう。
まず，クリアスクリーンが必要ない。ポリゴン表示では，これだけでも，もの凄く違う。と，いうのは実際に処理系を作ってみた人でないとわからないだろう。さらにBLITTERはCPUと並列に動作する。CPUは計算だけに専念できる。BLITTERはAGNUSの1機構であり，AGNUSとは26チャンネルの高速DMAそのものだといえばAMIGAの異様なパワーが理解できるだろうか。
SLASHの開発を傍らで見ていた者としていえば，フライトシミュレータを目指してそういうハードウェアに至った背景というのが容易に想像できる。非力なハードウェア（多分，ATARI800）を駆り，血と汗を流して3Dプログラムを開発したことのある人でなければ作れない機構なのだ。高速浮動小数点演算プロセッサとか矩形転送/台形転送機能とか3Dスクロール機能といった，強力だが用途の限られた機能に走らないところが素晴らしい。
AMIGA礼賛をやってもしかたないのだが，ツボを心得たハードであること以上に，実際にはソフト屋が凄い。
以前，横内君にAMIGAで有名なF1ゲームの逆アセンブルファイルを渡したことがある。しかし同じ68000CPUとはいえ，まったくの異文化であり，ハードウェアに関する資料が少なすぎて結局ほとんどなにもわかるはずがなかった。わかったことはカスタムチップの使い方は超絶的だが，それ以外のコードの最適化はたいしたことはない(現在のX68000レベルからすれば，だが)という衝撃の事実だった。
コーディングよりもアルゴリズムで稼ぐというのは理想だが，そうそう実践できるものではない。ソフトウェア技術での日英格差がまだ5年分くらいはありそうな感じだ。わずか7MHzの68000。少なくともRED ZONEクラスならひけはとらないはずだと思うのだが，とても同じことはできそうにない。そこにはやはりソフトウェア技術の壁がさらにぶ厚く存在するようだ。

そしてSLASH

X68000でまともなポリゴンゲームを実現するにはどうすればいいのだろうか？
ハードウェアでは多少不利だが，おそらくまともなアルゴリズムとそれなりのノウハウがあれば，そこそこのことはできるはずだ。とりあえず，現状でできるところまでを「ちゃんと」作ってみることが必要になるだろう。
抜ける手はすべて抜いて，ごまかせるものはごまかす。真面目で正直で安全に作られた3Dシステムには用はない。
そこでSLASHである。現在X68000でできる部分を突き詰めてまとめたものがこれだ。本当に重要なのはSLASHの上にかぶさる部分なのだが，システム化するにはまだノウハウが足りない。
結局のところ，このシステムは横内君が我流で作っているものだ。いくつかのサンプルを除けば，実際に運用されたのは付録ディスクの締め切り間際に作成されたSION Ⅳのデモだけといっていい。
3D関係のノウハウはこれから蓄積されていくことになる。現在までにもいくつかのアルゴリズムが試され，それらのなかでとりあえず現在のものが採用された。もちろん，処理と場合によってはほかのアルゴリズムのほうが効率よいということもありうる。さらによいアルゴリズムを知っている人だっているかもしれない。コーディングの無駄があるかもしれない。
現状では理論派丹明彦と職人横内威至のコンビがプロジェクトを進めているが，SLASHはもっと強力になるべきである。だから協力者を広く募集している。3D野郎はぜひ腕を貸してほしい。

＊　＊　＊

ここでSLASHの特徴をまとめてみよう。

●ポリゴンソートをしない

まず，SLASHでは物体内では面の前後判定を行わない。ポリゴンを定義された順番に上書きしていくだけである。裏を向いた面は描かれないのでうまく指定すればソートはほとんど必要なくなる。しかし，これには形状と視点との角度によっては絶対に破綻する場合というのがある。
SLASHでは形状を分割・マクロ化してソートすることでこれを回避できるように設計されているが，これはスッパリあきらめるほうがむしろ正しい。動いていれば，まずバレることはない。
視点や運動に極端な制限を加えたスターブレードでもポリゴンの前後関係が破綻することがあるのだが，普通の人はまず気づかない。知覚されないものは存在しないのと同じである。
問題はむしろモデリングに手間がかかるということだろう。しかし，きちんとモデリングされた物体ならなにも考えなくても高速表示が可能になるのだ。

●シェーディング対応である

頭に「スムース」の文字が入らないのが残念だが，要するに光源と面の角度によって明るさが変化するということだ。
リアルタイムゲームでシェーディングが採用されている例は驚くほど少ない。スターブレードなどでも行っていない。しかし，これは物体の表現力を非常に高めることができるのだ。モデリングも省力化できる。

●座標系

基本的な考え方はMAGICと似ているが軸の取り方がなぜか違っている。
2軸以上の回転で破綻するのはMAGICと同様である。現在，オイラー角の採用などが検討されているが，なにかうまい手があるのかもしれない。このあたりのノウハウは猛烈に不足している。

●もっと軽くする

時間的な制約から今回付録に収録されたSLASHは未完成(というかサブセット)である。テキストVRAMに対応した部分であるTSLASHが丸ごと抜けているのだ(途中バージョンまでは対応していた)。
すべての表示をグラフィックで行うのはやはり重い。テキストとの併用が効果を発揮することはすでに確認されている。
SLASHでは，さらにどうしようもない場合のためにラスター抜き表示ルーチンが用意されている。実測すると，ラスター抜きによって約30%の高速化が可能になる。いくらX68030が出たからといって，10MHz時の動作が基本だから，これはもはや標準装備ともいえるかもしれない。

なんのためのポリゴンか？

飛行機モノであれば自由に飛びたい，レースであれば，よりリアルに走りたい。これが基本的な欲求となる。我々はこのためにこそポリゴンを導入しようとしているのだから。
そういった意味では国産のポリゴンゲームは不満が残る。あれだけのシステムを持ちながら自由に動けないスターブレード，ポリゴンの魅力を十分に出しながら，ボロボロの走りしかできないバーチャレーシング。シルフィードに至っては，もはや誰もポリゴンゲームだとは思っていない。
こういったことははっきりいって表示そのものより数段難しいことなのだ。
かくして，机の上に自動車工学や航空力学の本が山積みされていくことになる。ああ，目指す山はあくまで高い。しかし，とりあえず登り始めなければなにも始まらないし，すでに歩き出してしまったのだ。後戻りはできない。

ハイテンションポリゴナイザライブラリ

SLASH ver.1.0

Yokouchi Takeshi 横内 威至

シェーディングまでもサポートした驚異のポリゴナイザライブラリ「SLASH」付録ディスクにあるサンプルから，そのすごさを実感してもらえたでしょうか まずは，その制作背景とリファレンスを紹介していきます

スタッフに横内威至という奴が2人いるんですか，なんて質問はしないでくれ。最近ゲームレビューしか仕事がなかったから，実は俺もプログラムが組めますよ，ってことを忘れていた人もいるに違いない。偉そうなことといってなにもできないカス，そんな汚名を晴らすためにも，少しはまともなこともやってやるぜ。

ハイテンションポリゴナイザライブラリ「SLASH ver.1.0」，なんの略語かは俺も決めていない。辞書を引くと「深く切る」「めった切り」「一撃」。けっこういけると思ったところ，「酷評」。まあいいかもしれない，ところが「ションベン」。

それはそうと現状では結構高速，シェーディングもサポートしたハイクオリティなシステムだし，使いたい人には使えるようにしてみた。目指すはDōGAをリアルタイムで動かしたり，グラフィックワークステーションを酷評するレベルまで？

怒りのコーディング

スタートは1月であった。AMIGAのポリゴンドライビングシミュレータを目前に，俺はアメリカ人に完全な敗北を感じていた。誰か素晴らしい作品をX68000で作っていないものだろうか。最初はまったくあきらめていた。なんといっても俺にはノウハウがまったくない。突然作ろうにもなにをどうするべきか予想もつかない。

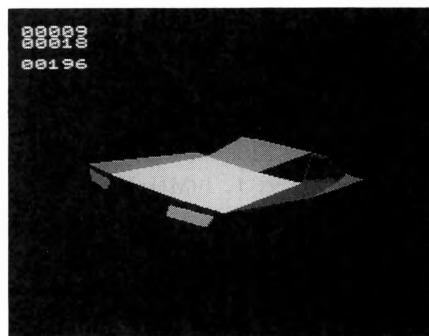
時がたって3月下旬，やっぱり俺の力でなんとかやってみようと思い立つ。とりあえず高速な四角形ルーチンでも作れればあとはなんとかなるかもしれない。と，基本的な3D処理，およびポリゴン処理ルーチンは3日で作り上げた。ver.1.0に仕立て上げるのには，1千時間におよぶ半牢獄生活を送ってしまったが。

最初は，なんだ，やってみればたいしたことではないんだなと思った。それと同時にポリゴンというのがそんなに重い処理ではないことに気づく。どういうことだ？世間に出回るポリゴンモノというのはかなり馬鹿なことしかやっていないのだろうか？　いろいろ解析したりして，その後時間をつぶす。確かに安易なルーチンも多い。本当に考えて作っているのだろうか。

よく考えてみると実は短絡的な概念を耳にすることがかなり多いことに気づく。たとえばポリゴンは絶対にテキストが速い，エッジ検出にはブレゼンハムの線分アルゴリズムが当然であるなど。「SLASH」が決してベストではないことをあらかじめ断っておく。だが少しは自分でより新しい方法を考えろ。ソフトは常に進化してゆく。失敗を繰り返してでもより新しいものを作れ。俺は世間にケンカを売る。読者すべてが俺のいうことに怒ってもかまわない。アグレッシヴに闘えば，いつか究め尽くしたモノが出てくることを信じている。

内容の説明

ある程度の限界はあるが，かなりのレベルでリアルタイム制御を目指して作った。そして誰も手を染めなかったシェーディングなんかもやってしまっている。シェーディングなんて重くて遅いから外してもっと速いヤツを作れ，って人は黙りなさい。世にも変態的なクレイジーシェーディングにより計算時間ほとんどなし状態である。だから，いやでもシェーディングさせて使うのが望ましいのである。ゴルゴも眉をひそめるほどびっくりモノの効果があるはずだ。

さて，実際サンプルを動かしてみればわかるけど，ポリゴンの描画でウソをつくことがある。見えない部分が見えることがあるのだ。なぜかといえば基本的に物体の面ソートをいっさいカットしているからだ。これはかなりのポリゴナイザが暗黙のうちに使っている技術(手抜き)のひとつである。

サンプルのテスタロッサ

スタークルーザーやスターフォックスを参照すればウソがわかる(違っていたら申しわけない)。ま，素人はこのことを知ってはいけないのだが。

しかし，本来はそうだが，対策としてとてもいい加減なマクロソート(仮)を導入してある。物体のポリゴンを複数ブロックに分割し，指定した参照点によってソートを行うというとても邪道なソートをサポートした。これによってこの致命的であると思われたソートなしポリゴンは，一気に立派なポリゴナイザへと発展した。かなり複雑な形状である飛行機などの物体でも，問題なく表示が可能である。このあたりの内部概念についてはここでは説明しない。今月から始まる丹氏と俺の暴力的連載「ハードコア3Dエクスタシー」で，あらゆる技術を培っていきたい。

ASLASH&GSLASH

「SLASH ver.1.0」はライブラリとして提供されている。最初はソースレベルで提供し，使いたい人間が好きにいじれるようにするつもりであった。そうでないとなれば当然手続きなどで，わずかなロスが出ることになる。基本的にはリンクして直接コールできるため，一部のサブルーチン以外はロスなしといえるが。

また，表示に関してはかつて8ビット版

より実践的なサンプルSION Ⅳ(DEMO)

スタークルーザーで行っていた，ラスター抜きもサポートしている。これは，SLASHLIB.Aの前にSLASHRLIB.Aをリンクすれば，ラスター抜きポリゴン描画モードとなる。現段階ではスイッチによる指定はできない。以下に説明する不都合により，実行ファイル上に同時に存在することが不可能だからだ。これは，10MHz機種ではかなり有効な手抜きである。CPUがポリゴン描画，3D計算のすべてを行わねばならないので，この莫大な表示はかなりの負担である。この表示を半分に減らすだけで何十%かは処理を省けるのである。

そして，アセンブラプログラマは以下の内容は把握しておくことが望ましい。

ライブラリの中身で直接参照できるオブジェクトは基本的に2つ。ASLASHとGSLASHの2つ。あとは4096段階のサインテーブルも参照できる。

・ASLASH.O：アフィン変換による座標計算を主に行う。

・GSLASH.O：グラフィック全般の処理を行う。ポリゴナイザとしては本体にあたる部分。

そのほかはサインテーブルのオブジェクト，そしてシェーディング用のデータテーブルである。このデータが300Kバイトにもおよぶため，リンクするにはちょっとした操作が必要となる。GSLASHからASLASHを参照する部分があり，これが絶対アドレス指定ではなくプログラムカウンタ相対を使用するため，ASLASH，GSLASHの順番に並べなくてはリンクできない。これは制作するプログラムが外部参照するシンボルの出てきた順にリンクするためである。間違った使い方をしなければ気にする必要はない。

Cプログラマの方は丹氏による解説を参照してもらうことにする。BASIC，そのほかの言語に関してはサポートされていないが，要望しだいでいくらでもサポートしようとも思う。

データフォーマット

データフォーマットの概要は表1に，SLASHのファンクションコールを表2に示す。より深く知らねばならないことについては以下に説明を加える。

●座標値について

「SLASH」で扱える座標は符号つきワード（−32768〜32767）である。ただし計算上，ワード境界をオーバーすることがあるため，限界までの値を使うのは避けること。精度が悪いと考えるかもしれないが，このシステムはDōGAとはまったく別のリアルタイム制御システムであるため，高速化の犠牲となっているものは非常に多いことを覚えておくこと。

さらに追い打ちをかける仕様として，画面手前Zクリッピングにより切られる平面での座標値が16倍として扱われているため，−2048〜2047の範囲でしか本当に正しい表示が行われない。

しかもこれに計算が入るため，さらに小さい値でモデリングすることが望ましい。推奨座標値は−512〜511までと，一応恐れてかなりせばめようと思う。これは計算が内部でどうなっているかを俺自身が完全に把握すれば完全な値を設定できるが，なんにせよどのような状態かを想定することが完璧に行えないため，現状ではっきりといい切らないことにする。まあ，「SLASH」は計算上理にかなわない値がくる可能性を考え，かなりタフには作ってあるのであまり心配はしなくてよい。

画面手前での倍率によって調整できないのか，と考えるかもしれない。だが画面手前でもし1倍なら，モデリングは結局大きくしなければならない。となれば同じように限界値はくるし，そのほかにもいろいろと深い問題があるためさほど変わらないようだ。これは以降のバージョンでも変化させる可能性はあるので，モデリングデータはしっかりと保管することを要求しておく。

●頂点について

頂点数については物体ひとつにつき2048個までと限定している。これについては十分実用レベルであり，問題ないと思う。ポリゴンリストで指定する頂点番号は，0〜N-1であることに注意してほしい。

また，変換に使うワークは頂点1個につき16バイト必要。もちろんワークは，物体ごとに指定する。

●変換，表示，クリアについて

ワークとしてさらにグラフィッククリアを主な目的とした，変換後の最大最小座標ワークエリアを必要とする。これで指定された矩形領域を何個か連続に高速クリアするルーチンを用意してある。これは物体が十分小さいときに画面をオールクリアするのは無駄であること，またX68000のパワーとのバランスを考えたうえで用意したものである。もちろん描画面積が大きかったりすればオールクリアが有効な場合もあるが，そのへんの管理はユーザーが行うべきである。

さて，このクリアルーチン関係で変換，表示，クリアの動作がかなり複雑なのでさらに説明する。

1)　まずワークを指定してTRANSLATERで座標変換を行う。そして，これによって指定されたワークに最大最小値を書き込む。このときZクリッピングに引っ掛かる座標についてはまだ計算に入らないため，この段階では最大最小値は意味をもたない。

2)　続いて同じワークを指定してDRAWPOLYにより描画を指示する。変換に失敗した座標を含むポリゴンがあればエッジとクリッピング平面との交点を求め，その値によってワークを変更する。これを全ポリゴンについて行う。

3)　最後に同じワークを指定してADJUSTMINMAXをコールすると，これらの最大最小値を設定した表示領域でクリッピングする。このとき，わずかでも表示領域内に入っていればこれらの値を修正し，ポインタはワーク分（8バイト）インクリメント，そしてそこにエンドコードを書き込んでリターンする。もしまったく表示領域内に入っていなければこのワークの先頭にいきなりエンドコードを書き込んでリターン。

さらに続けて物体を変換，描画するならこの段階で返ってきたアドレスをワークとして指示すれば，物体の数に左右されない画面単位でのクリア情報が作られるのである。

さて，今度は表示だが，一般的な使い方としては当然ページを2枚（領域を2つ）使って交互に表示する，という方法で対応することになる。このために「SLASH」では画面左上にあたるアドレスを任意に設定できる。ただしエラーチェックはしていないため，下手なアドレスを指定すると暴走する。そしてこのページ切り替えを「SLASH」ではいっさい制御していない。これはユーザーが独自に扱うこと。当然ミニマックスワークは使用ページ分用意する必要もあり，あらゆる動作を自分で管理できなければ使用不可能である。

●ポリゴンについて

ポリゴン数について制約はない。ワードで表せる最大値まで可能のはず。実用レベルからみれば，物体ひとつで100ポリゴンぐらいがいいところかもしれない。これは直接モデリングに関わることであり，センスしだいである程度簡単に抑えることができよう。サンプルでつけたテスタロッサはあまりよくないモデリングである。

ポリゴンを描画するときは与えられたデータ順に処理する。基本的には遠くの物体から描画させるのが正しい使い方であるが，これも「SLASH」ではいっさい管理しない。これはユーザーが管理したほうがはるかに効率がよいのである。こうなると問題が出てくるのが物体単位での面判定である。これは確実に不都合のあるモデルもあるが，ある程度，データを作る段階で処理できることである。なるべく不都合の生じないように気をつけてモデリングしなくてはならない。そしてどうしても不可能な判定のために部分ソートを行うのである。これでほとんどの物体は問題なくいくはずである。

また，四角形は基本的には同一平面上でなければならない。しかしどんな精度をもっても，計算による誤差が出るため変換後に同一平面上にはならないことがある。「SLASH」ではこの不都合を吸収するためどんな4点でも面として表示できる。ただし狙ったとおりの面にならない可能性は大きい。あまり酷いデータは与えないように心がけること。

ラインも形ばかりであるが指定できる。これは任意の頂点で形成される三角形上の2頂点を結ぶ線分として扱う。これにより線でも表裏の判定が行われており，ポリゴンと同レベルでの扱いが可能。法線の方向はもちろんこの三角形の法線に等しい。

また点も指定可能。任意の4頂点で四角形を指定し，その面上の4頂点として扱われる。頂点をマイナスの値にすればその点は表示されない。また内部では外積による表裏判定で4点目は含まれていないため，4点目だけはあまり正しい表示がされない可能性がある。だがあまり気にするようなものではないので無視することにする。

●カラーテーブルについて

パレットコードは本来16ビットである。ただし，内部での処理を軽減させるために1色を1ロングワードで管理している。上位ワードと下位ワードを同じ値にすること。別々の値にすればメッシュらしくなるが，正しくサポートしていないため禁止。シェーディングによる色管理のほうがはるかに効果があるためメッシュは無視する。テーブルはシェーディングのために1色あたり32個，128バイトのデータを用意すること。サンプルで示すとおり，うまく設定すれば素晴らしい光沢の表現が可能。ハイライト指定や艶消し，環境光の設定もユーザーのセンスしだいである。もちろんこのシェーディングを使うかどうかは面単位で設定する。使用しないならばパレットテーブルアドレスはそのままパレットコードを置くことになる。

●クレイジーシェーディングについて

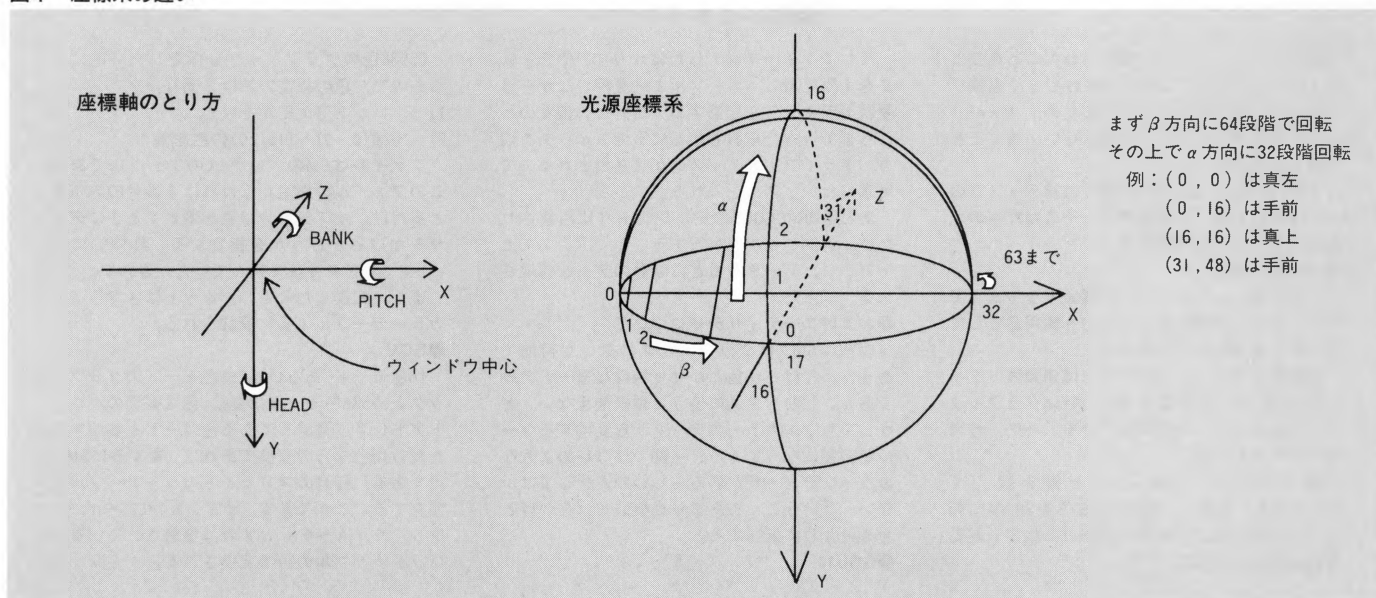
これはあらかじめ規定にそった法線情報を各ポリゴンがもたねばならない。また光源は物体ごとに指定することが可能。幸いこれは「SLASH」がサービスコールとして自動設定ルーチンを設けてある。光源と面での指定座標は系が違うため，別の座標系で考えてほしい。サービスコールのおかげで，特殊な用途以外でユーザーが管理するのは光源系だけでよい。具体的な値がどういう意味をもっているかは図1を参照すること。もしこのシェーディングがよくわからなければ面の法線αを－1に設定し，単色で動かしてもらいたい。この光源に関するテクニックは連載で扱うことにする。

今後の拡張

まずバグについてだが，おそらく確実に存在するはず。なんといっても計算が複雑であるため，どこで異常な値を扱っているかはまだ不明である。ある程度いじめても動いているため，かなりのバグは取れているはずであるがバグは永遠にプログラムにつくものであり，実際使いこなさねば見えてこない。もし異常を感じたらぜひOh!X編集部まで連絡を願してほしい。またこのソフトを使用したうえでの支障，損害に関して我々はいっさいの責任をもたない。

バージョンアップとしては当然このバグフィックス，テキスト版ももちろん，そして68030，68040(?)，もしかしてRISC版(?)，POLYPHONやV70なども考えられる。もし私が手に入れたなら，でもあるが。

なお，68030ではもしかしたらキャッシュ絡みで動かないかもしれない（現在のところ動作はしている）。当然自己書き換えがあるからだ。68030なのにもったいないかもしれないが，68030専用コードで書かれていないこと自体すでにもったいないので無視し

図1　座標系の違い

ている。

また，68030で劇的なスピードが得られないのは，グラフィックはクロックが上がっていないからである。これは俺が68030の魅力を半減させるかなりの原因であると思っている。まあ，でも16MHzよりもかなり高速であり，十分感動できる。

また，さらに加速してスムースシェーディングやマッピング，完全ソートなんかも考えてみたい。現状ではこれらはかなり苦しいレベルであるのだが。

ライセンスについて

「SLASH」は完全なフリーソフトにはなっていない。しかし，個人使用に関しては制限などつけない。ただし移植，そのほかに関しては連絡をしてもらいたい。

また同人ソフトの域を超えた商業使用，利益目的の使用は別途契約を必要とする。具体的にわからない点があれば連絡をしていただきたい。これはいずれ納得できる条件になれば制限を外すことになると思う。

あとがき

元々バージョンは2つあった。だから「SLASH ver.3.0」が正しいかもしれない。それぞれアルゴリズムがすべて違っている。どの方法が最も効率がよいかチェックしながら3本同時進行であった。ほかになにがあったかというと，画面手前から描画して描画の無駄を完全に省くヤツ，そしてリアルタイム完全Zバッファバケットソートスキャンなんていうバグがどうしても取れなかったバージョンである。

だが今回発表したものがX68000では最も優れたスピードを叩き出した。これが別のハードとなれば以前のバージョンが有効となるかもしれない。しかしアレだよな。こんなものは現在どこでもハードウェアでやってることだよな。某DOS関係マシンでさえハードで簡易ポリゴン描画があるんだぜ。もし次のマシンで納得できることができなければ俺はもうやめるぜ，いい加減。積むだけ積んでまったく使われていないチップ，っていうものに無駄な金をかけているというのはやはりアレだよ。

一応，なんとか「SLASH ver.1.0」として完成したが，いったいどれほどの人間が扱えるのだろうか。すでにかなりのユーザーはシャットアウトしてしまったかもしれない。また，ポリゴナイザがあるからといって簡単に3D処理が行えるわけではない。このあたりの技術は連載のほうで磨くことになるであろう。ぜひ，ひとりでも多くの人間が3D，ヴァーチャルドラッグで体を蝕むことを願っている。

さすがに今回は疲れはててしまった。2カ月間ほとんどをこの開発に費やしていたからだ。寝ても起きてもクロックを削り，そしていつまでたっても続く怒濤の拡張，泣ける嵐のような処理展開，大きくなればなるほど繁殖する永遠の恋人のバグ。そして自分のシステムをいつまでたっても使えず美味しい外周りを自分で開発できないこの苦痛。最終的には汎用性をもたせるためにそれなりにパワーダウンしてはいるが，なにも例外処理をしていない初期バージョンのスピードに追いつけるよう死力を尽くした。

現段階ではもうほとんど限界に近いスピードを叩き出していると思う。動けばよい，なんて甘ったるい精神ではなく限りなく効率を追求し，うまくレジスタを誘導してスタックなどの比較的遅い命令は排除する。同じようなルーチンでも場合によってはほんのわずかな違いが出てくるためマクロも使えない。1クロックでもロスが見えていれば展開して専用ルーチンとして書き直す。少なくとも全体をそうするつもりではあった。これが究極の速度を出しているとは信じない。なんといっても体力切れで手が回っていない部分もあるのだ。そしてスタークルーザーとどちらが速いか，といってもまだ完全にはわからない。場合によっては負ける可能性もあるだろう。だがこれはこれで完成したプログラムである。もし劇的に速いものを作るとすれば，それはアルゴリズム自体を変えなければならないはずである。ウソだったらごめん。

こんな開発状況だから，もう生活は非人間的。とにかくより速いルーチンにするため，丸3日考え数時間で手直し，そして1週間かけてデバッグ＋拡張，こんなことを死亡寸前まで繰り返す。このオモテに表れない，目に見えない苦労は誰にもわからない。休息さえ許されない。突っ走らねばコードはすべて頭から消えてしまうのだ。

それと競輪のほう，ごめん。全国に棲息する栄光の愚か者から苦情のハガキが……。忙しくて結局手がまわらなかった。いつかチャンスがあったらやらしてもらいたい，と弁解しておく。あ，やっぱりポリゴンでかな。

追記&おまけツール

追記として，本文中に書き忘れたことをひとつ。SLASHの内部ではなんの操作もなく直接グラフィックなどをアクセスするため，スーパーバイザモードでなければ動作しない。清く正しく突っ走ってほしい。

あと，SLASHには関係ないが，付録ディスクの中に収録したツールで説明をしそこねたものがあるので，以下に解説をする。

●DRACLOCK

これはなにかというと，悪魔城ドラキュラでプレイヤーを感嘆させた時計台を実用品として使ってしまおうというものだ。

基本的なグラフィックデータは悪魔城ドラキュラのデータをもとに変換し，針はグラフィックの回転ルーチンで描画している。一応，秒針まで加えておいた。

作り方だが，CONFIG.SYSを書き換えてSRAMDISKを登録し，SAVESC.SYSをSRAMに転送。次にSWITCH.XでBOOTをRAM1に指定し起動をSRAMからにする。

そしてフォーマットしたばかりの2HDディスクを1枚用意し，カーソルキーを押しながら悪魔城ドラキュラを起動する。時計台の狼女のところまで行ったらおもむろに電源スイッチを切る。ドライブ1のディスクが吐き出されるので，用意したディスクを入れる。

あとはDRACLOCKのディレクトリに用意されたバッチファイルを実行することで，ここでセーブされたデータをもとに時計のデモが作成される。以上だ。

●おまけユーティリティについて

以前のSM.Xはグラフィックの扱いが可能であった。だけど16色とかなり特殊なモードのみである。だから普通に使うと結構使えない。どうしてもフルカラーのデータから変換するツールが必要になってくる。一部ではコレのようなカラーレデューサがあるらしいが浸透してはいない。ということであまり高機能ではないけどおまけとして配布する。

●GCV16

65536色のグラフィックを指定した16色にするもので，色の指定方法は2通り。

1) パレットファイルを読む。
2) 座標(0, 0)～(15, 0)の色を使う。

ファイルはSMの.PAL形式のファイルである。このファイルの指定がなければ2番目の方法をとるだけ。オプションは色を落とすときにディザをかけるかどうかを指定する。具体的には"－?"オプションで読んでいただきたい。

また，変換したときのパレットはスプライトカラーテーブル1番に登録される。

●GCV

16色モード，あるいは256色モードのグラフィックを65536色モードに変換。色は実際のパレットアドレスに登録してある色コードを参照する。ただし16色からの変換であれば（要するにSM.X用である）任意のスプライトカラーテーブルを指定することができる。オプションに"－n"（nは0～9，A～F）とすれば登録されているパレットテーブルの色で変換される。

[表1　データフォーマット]

[パラメータ] ――――――――PARAMETER

00	x.w	物体X座標
02	y.w	物体Y座標
04	z.w	物体Z座標
06	pitch.w	PITCH(X軸回りの回転角度)
08	head.w	HEAD(Y軸回りの回転角度)
10	bank.w	BANK(Z軸回りの回転角度)
12	α.w	光源緯度(0～31)
14	β.w	光源経度(0～63)

[ポリゴンマクロ] ――――――――POLY. MACRO

n.w	ブロック数(2, 3)
point1.w	ブロック1の重心(ポイントリストの番号)
point2.w	ブロック2の重心(ポイントリストの番号)
point3.w	ブロック3の重心(ポイントリストの番号) (ブロック数3のみ必要)
POLYLIST1.l	ブロック1のポリゴンリスト
POLYLIST2.l	ブロック2のポリゴンリスト
(POLYLIST3.l	ブロック3のポリゴンリスト)

・物体を指定したブロックに分割し，それぞれの重心（参照点）を基準にしてソートをさせるために使用するデータ。現段階では，最高ブロック数は3ブロックまでである。

[ポリゴンリスト] ――――――――POLY. LIST

n.w　面数

ポリゴン1枚につき16ワード。以降は各ポリゴンヘッドからのオフセット値で示す。

[

00	type.w	識別子 0:三角形　1:四角形 2:線分　3:ポイント
02	point1.w	頂点1 (ポイントリストの番号0～2047)
04	point2.w	頂点2
06	point3.w	頂点3
08	point4.w	頂点4
10	α.w	法線緯度(－1, 0～31)
12	β.w	法線経度(0～63)
14	color.l	カラーテーブルアドレス，あるいはパレットコード
16～31		リザーブエリア

]

・以上を面数分だけ用意する。頂点は1～4に向けて右回りが可視面となる。

・三角形，線分のとき，頂点4は無効。

・線分のとき描かれるのは，頂点1，2を結ぶ直線。頂点3は可視判定用。

・ポイントのときは頂点1～4で構成される面上の点1～4の点のみを描画。ただし頂点番号をマイナスで指定するとその点は描画されない。マイナス指定は，[NOT（ポイント番号）]である。

たとえば頂点1がポイント0，頂点3がポイント5でそれぞれを描画なしとすると，頂点1は－1，頂点3は－5を設定する。

・色はカラーテーブルがあるアドレスの先頭を指定する。ただし法線緯度がマイナス指定（拡張のため一応－1にすること）のときは，直接色コードを指定する。色コードは同じコードを2つ並べてロングワードにしておくこと。

・リザーブは拡張用。ユーザーが自由に使用可能。ただし，今後のバージョンで使う可能性もあるので，後ろから使うのが望ましい。

[ポイントリスト] ――――――――POINT LIST

n.w　頂点数

ポイント1点につき6ワード。以降は各ポイントヘッドからのオフセット値で示す。

[

00	x.w	頂点X座標
02	y.w	頂点Y座標
04	z.w	頂点Z座標

]

以上を頂点数分だけ用意する。

・点は最大2048個まで。ポイント番号は0～n－1となる。

[トランスレートワークエリア] ――――――――TRNS. WORK

ds.w　頂点数×8

頂点ひとつにつき16バイト必要。2D変換後，作成されるデータは次のとおり。

[

00	x.w	3D計算後のX
02	y.w	3D計算後のY
04	z.w	3D計算後のZ
06	mx.w	2D変換後のX(成功時)
08	my.w	2D変換後のY(成功時)
10	z.w	＋04と同じ(成功時)
12	0.w	
14	flg.w	0：2D変換成功，1：失敗

]

・2D変換はそのまま画面上での座標を示す。

・変換失敗はZ軸（画面に垂直な軸）に対するクリッピングにかかることを表す。

[ミニマックスワークエリア] ――――――――M.M. WORK

ds.w　物体数×4＋1

物体ひとつにつき8バイト＋2バイト（エンドコード用）必要。ワークの内容は以下のとおり。

[

00	Ymin.w	Y座標最小値
02	Ymax.w	Y座標最大値
04	Xmin.w	X座標最小値
06	Xmax.w	X座標最大値

]

これを画面に対して有効な物体数分だけ用意する。

8n＋0　$ffff.w　エンドコード

・画面クリアに対する情報なので，物体ごとにワークを分けることはせずに画面単位で分けることにしてある。

[カラーテーブル] ――――――――COLOR TABLE

[

00	Color0.l	光源と正反対の向きでのカラーコード
04	Color1.l	
08	Color2.l	
⋮		
124	Color31.l	光源の向きでのカラーコード

]

・1色分で32段階のテーブルが必要。

[表2　ポリゴナイザラブライリ SLASHファンクションコール]

●TRANSLATER1

Ct.　：ASLASH

IN　：A6.L＝PARAMETER
(3D変換パラメータ先頭アドレス)
A4.L＝1POINT LIST
(1点分のデータの先頭アドレス)

OUT：D0.w＝回転後のZ座標
D1.w＝0(2D座標有効)
－1(2D座標無効)
D2.w＝回転後のX座標
D3.w＝回転後のY座標
D4.w＝2D変換後のX座標
D5.w＝2D変換後のY座標

BRK：D6, D7/A0, A1, A2, A6

CNT：A4

・2D座標無効は，3D変換した点がZ方向クリッピング平面より手前に

入る。
・IPOINT LISTは以下のとおり。

```
[
+00    X.w
+02    Y.w
+04    Z.w
]
```

・CNTは以降継続を意味する。よってA4はコール時のアドレス+6を示す。

●TRANSLATER

Ct.　: ASLASH
IN　: A6.L=PARAMETER
　　　　(3D変換パラメータ先頭アドレス)
　　　A5.L=TRNS. WORK
　　　　(トランスレートワーク先頭アドレス)
　　　A4.L=POINT LIST
　　　　(ポイントリスト先頭アドレス)
　　　A3.L=M.M. WORK
　　　　(ミニマックスワーク先頭アドレス)
OUT : なし
BRK : D0〜D7,A0, A1, A2, A6
CNT : A3, A4, A5
・パラメータに従って頂点リストを2D変換し，トランスレートワークに2D変換リストを作る。またその点リストのX，Y各最大最小値をミニマックスワークに書き込む。
・A3はコール時のアドレス+8を示す。この段階でエンドコードは書き込まれない。
・A4=POINT LIST+頂点数×3ワード+1ワードを示す。
・A5=TRNS. WORK+頂点数×8ワード+1ワードを示す。
・ポイントリストに1点もない場合の最大最小は，それぞれ+32767，−32768が書き込まれる。

●TRANSLATEREP

Ct.　: ASLASH
IN　: A6.L=PARAMETER
　　　　(3D変換パラメータ先頭アドレス)
　　　A5.L=TRNS. WORK
　　　　(トランスレートワーク先頭アドレス)
　　　A4.L=POINT LIST
　　　　(ポイントリスト先頭アドレス)
　　　A3.L=M.M. WORK
　　　　(ミニマックスワーク先頭アドレス)
OUT : なし
BRK : D0〜D7/A0, A1, A2, A6
CNT : A3, A4, A5
・TRANSLATERと同じものだが，回転パラメータを参照しない。すでにTRANSLATER(TRANSLATER1は無効)をコールしてあり，同じパラメータで次の物体を変換するときにこれを使用する。内部で同じ行列を作ることは無駄なのでこのコールを用意してある。
・パラメータに従って頂点リストを2D変換し，トランスレートワークに2D変換リストを作る。また，その点リストのX，Y各最大最小値をミニマックスワークに書き込む。
・A3はコール時のアドレス+8を示す。この段階ではエンドコードは書き込まれない。
・A4=POINT LIST+頂点数×3ワード+1ワードを示す。
・A5=TRNS. WORK+頂点数×8ワード+1ワードを示す。

●ADJUSTMINIMAX

Ct.　: ASLASH
IN　: A0.L=M.M. WORK
　　　　(ミニマックスワーク先頭アドレス)
OUT : D0.w=0(画面範囲内に点がかかる)
　　　　−1(画面範囲内に1点もない)
　　　A0.l=継続して使えるワーク先頭アドレス
BRK : d1,d2
CNT : --
・変換，描画終了後，設定されたウィンドウに合わせて，ミニマックスワークに書き込まれた座標最大最小値を更新する。
・画面範囲内に1点でもあれば，設定されたウィンドウサイズに最大最小値を合わせる。A0はコール時のA0+8を示し，そこにエンドコード(−1)を書き込む。
・画面内に1点も含まれないとき，A0はコール時のアドレスのままで，そこにエンドコードを書き込んでリターンする。

●DRAWPOLY

Ct.　: GSLASH
IN　: A6.L=POLY. LIST
　　　　(ポリゴンリスト先頭アドレス)
　　　A5.L=TRNS. WORK
　　　　(トランスレートワーク先頭アドレス)
　　　A4.L=M.M. WORK
　　　　(ミニマックスワーク先頭アドレス)
OUT : なし
BRK : D0〜D7, A0〜A4
CNT : A6
・ポリゴンリスト，トランスレートワークに従ってポリゴンを描画をする。
・Z方向手前のクリッピングにかかるポリゴンがあれば指定されたミニマックスワークを更新する。この段階ではエンドコードは書き込まれない。

●DRAWPOLYSORT

Ct.　: GSLASH
IN　: A6.L=POLY. MACRO
　　　　(ポリゴンマクロ先頭アドレス)
　　　A5.L=TRNS. WORK
　　　　(トランスレートワーク先頭アドレス)
　　　A4.L=M.M. WORK
　　　　(ミニマックスワーク先頭アドレス)
OUT : なし
BRK : D0〜D7, A0〜A4, A6
CNT : ——
・ポリゴンブロックで示される重心点によって複数のポリゴンリストの前後を判定し，そのうえで表示を行う。

●CLEARBOX

Ct.　: GSLASH
IN　: A0.L=M.M. WORK
　　　　(ミニマックスワーク先頭アドレス)
OUT : なし
BRK : D0〜D7, A1〜A6
CNT : A0
・ミニマックスワークの内容に従って画面を消去する。
・エンドコードを見つけるまで複数の矩形クリアを実行する。

●SETCCL

Ct.　: SLASH
IN　: D0.L=COLOR CODE　　画面背景色
OUT : なし
BRK : なし
CNT : --
・クリアするときのカラーコードを設定する。基本的には0を指定する。

●SETWPLN

Ct.　: GSLASH
IN　: A0.L=GRAPH. ADDRESS
　　　　(画面左上にあたるアドレス)
OUT : なし
BRK : なし
CNT : --
・描画する領域の最も左上に相当するアドレスを指定する。アドレスのチェックはいっさい行わないので，危険な値で動かすと暴走は免れない。

●SETCPLN

Ct.　: GSLASH
IN　: A0.L=GRAPH. ADDRESS
　　　　(画面左上にあたるアドレス)
OUT : なし
BRK : なし
CNT : --

・消去する領域の最も左上に相当するアドレスを指定する。アドレスのチェックはいっさい行わないので，危険な値で動かすと暴走は免れない。

●SETWINDOW

Ct.　: GSLASH
IN　: D0.w＝X.w
　　　(描画ウィンドウ横ドット数)
　　D1.w＝Y.w
　　　(描画ウィンドウ縦ドット数)
OUT : なし
BRK : D0, D1
CNT : --
・指定するドット数は16～256とする。エラーチェックはいっさい行わないので各自注意すること。

●SETWINDOWCENTER

Ct.　: GSLASH
IN　: D0.w＝X.w
　　　(描画ウィンドウ消失点X座標)
　　D1.w＝Y.w
　　　(描画ウィンドウ消失点Y座標)
OUT : なし
BRK : D2, D3
CNT : --
・描画ウィンドウ左上を(0, 0)としたときの座標で設定する。基本的にはウィンドウ中央を指定する。極端な値を設定したときの画質の保証はしない。

●PSET

Ct.　: GSLASH
IN　: A6.L＝POINT LIST
　　　(ポイントヘッドアドレス)
　　A5.L＝TRNS. WORK
　　　(トランスレートワーク先頭アドレス)
　　A4.L＝M.M. WORK
　　　(ミニマックスワーク先頭アドレス)
OUT : なし
BRK : D0～D7, A0～A5
CNT : A6
・ポイントデータリストに従って点を表示する。
・A6は次のポリゴンヘッドを示してリターンする。

●LINER

Ct.　: GSLASH
IN　: A6.L＝LINE LIST(ラインヘッドアドレス)
　　A5.L＝TRNS. WORK
　　　(トランスレートワーク先頭アドレス)
　　A4.L＝M.M. WORK
　　　(ミニマックスワーク先頭アドレス)
OUT : なし
BRK : D0～D7, A0～A5
CNT : A6
・ポイントデータリストに従ってラインを表示。
・A6は次のポリゴンヘッドを示してリターンする。

●TRIANGLE

Ct.　: GSLASH
IN　: A6.L＝TRI. LIST (三角形ヘッドアドレス)
　　A5.L＝TRNS. WORK
　　　(トランスレートワーク先頭アドレス)
　　A4.L＝M.M. WORK
　　　(ミニマックスワーク先頭アドレス)
OUT : なし
BRK : D0～D7, A0～A5
CNT : A6
・ポイントデータリストに従って三角形を表示。
・A6は次のポリゴンヘッドを示してリターンする。

●TETRAGON

Ct.　: GSLASH
IN　: A6.L＝TET. LIST (四角形ヘッドアドレス)
　　A5.L＝TRNS. WORK
　　　(トランスレートワーク先頭アドレス)
　　A4.L＝M.M. WORK
　　　(ミニマックスワーク先頭アドレス)
OUT : なし
BRK : D0～D7, A0～A5
CNT : A6
・ポイントデータリストに従って四角形を表示。
・A6は次のポリゴンヘッドを示してリターンする。

●PSET2

Ct.　: GSLASH
IN　: 点XY座標：P1 (D0.w, D1.w)
　　　カラー：c (D2.w)
OUT : なし
BRK : D0, D1, A0
CNT : --
・レジスタの内容に従って点を表示する。

●LINER2

Ct.　: GSLASH
IN　:　始点：P1 (D6.w, D2.w)
　　　終点：P2 (A4.w, D7.w)
　　参照点：P3 (A3.w, D1.w)
　　　カラー：c (D0.w)
OUT : なし
BRK : D0～D7, A0～A6
CNT : --
・レジスタの内容に従ってラインを表示する。

●TRIANGLE2

Ct.　: GSLASH
IN　: 頂点1：P1 (D6.w, D2.w)
　　頂点2：P2 (A4.w, D7.w)
　　頂点3：P3 (A3.w, D1.w)
　　カラー：c (D0.w)
OUT : なし
BRK : D0～D7, A0～A6
CNT : --
・レジスタの内容に従って三角形を表示する。

●TETRAGON2

Ct.　: GSLASH
IN　: 頂点1：P1 (D6.w, D2.w)
　　頂点2：P2 (A4.w, D7.w)
　　頂点3：P3 (A3.w, D1.w)
　　頂点4：P4 (A2.w, D3.w)
　　カラー：c (D0.w)
OUT : なし
BRK : D0～D7, A0～A6
CNT : --
・レジスタの内容に従って四角形を表示する。

●ADDNORM

Ct.　: ASLASH
IN　: A6.w＝POLY. LIST
　　　(ポリゴンリスト先頭アドレス)
　　A5.w＝POINT LIST
　　　(ポイントリスト先頭アドレス)
OUT : なし
BRK : D0～D7, A0～A6
CNT : --
・ポリゴンのシェーディングに関する情報，面法線を自動計算してポリゴンリストを書き換える。

●GETSQR

Ct.　: ASLASH
IN　: D0.L＝数値 (符号なし32ビット整数)
OUT : D0.L＝SQR(D0.L)
BRK : なし
CNT : --
・引数の平方根を求める。完全整数演算のため速いが，精度はあまり保証できない。

SLASH開発支援ツール

SLASH用簡易モデラ

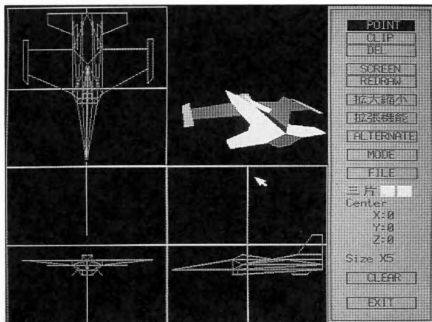
Kikuchi Isao 菊地 功

高性能なSLASHも扱うとなるとサポートツールが必須となります
なんといってもエディタがなければ始まりません
ここでは暫定版のエディタを紹介します

いきなりごめんなさい。実はこのモデラはまだ完成していないのです。そもそもこのモデラはMAGIC用に数年前に作られたもので，経験も浅かったことから（いまも経験豊富とはいえませんが）ソースはぐちゃぐちゃ，とにかく動けばいいやといった感がありました。

しかもあとからあとから機能を付け足し，自分でさえなにをやっているのかわからなくなっていたところに，U氏の「ポリゴン用に作り直してください」のひと言。いっそ最初から作り直そうかとも思いましたが，時間がないので却下。

ちょっとソースを眺めてみれば，そのうち思い出すだろうと楽観視していたのですが，どっこいさっぱりわからない。自分の作ったソースを見て頭を抱えて部屋を転げ回ってしまいました。もっともソースの汚さではいまも変わらないかもしれませんが。しかも作っている最中に肝心のポリゴナイザ自体のデータフォーマットが変わったりと，さんざんでした。

結局データフォーマットは変えずに以前のポリゴナイザをそのまま使うことにしましたけどね（吐き出すファイルは後述のアセンブラコードに変換するコンバータを通して最新版でそのまま使用できます)。まあ，どうにかここまでこぎつけて，なんとかモデリングはできるようになりました（やっぱり最初から作ったほうが早かったかもしれない)。

これが基本画面

こんな感じですからまだまだ山ほどバグがあり，使い方によっては，ときどき，しばしば，しょっちゅう腐ります。まじめにモデリングしようというときは，こまめにセーブすることをお勧めします。と，のっけからいいわけモードに入ってしまいましたが，そろそろ本筋に入りましょう。なお，各機能の動作安定性については，バグ出現度を参考にしてください。

使用法

いまさらですが，このモデラは横内氏制作のポリゴナイザで使用できるオブジェクトの作成を目的としています。ポリゴナイザ自体の機能の説明は横内氏の記事を読んでください。このモデラの特徴は，

1) マウスとキーボードの併用で効率的に作業が行える(かどうかはよくわからない)
2) 描画などの基本機能に加えて，移動・反転などの拡張機能を装備
3) ワークを2面持ち，裏画面の参照・合成などが可能（なハズなんだが)
4) 作成したオブジェクトの形状をその場で確認できる

と，こんな感じですが，まだバグだらけなので，あまり断言はできません。

また，注意ですが，ワークはそれぞれ128Kバイト確保してありますが，これを超えても警告など出さずに空きメモリを侵食します。データが128Kバイトを超えないようにしてください。まあ，ちょっとやそっとでは超えないでしょうし，128Kバイトを超えたデータがアニメーションできるかは疑問です。

それから，動作中にシステムエラーが起きたり，インタラプトキーが押されると，通常表示されるダイアログが表示されずにいきなり終了します。テキストVRAMや割り込みベクタの都合でこのようにしましたが，バスエラーなどが起きた場合は，どのみちデータの復旧は不可能になっているでしょうから，諦めてください。ただし，挿入されていないMOにアクセスした場合，バスエラーが起こることがあるようですので，その点は注意してください。

起動方法ですが，オプションなどはなく，コマンドラインからただ，

MODEL＜リターン＞

と入力するだけです。起動すると図1のような画面が現れます。

・平面図/正面図/側面図

基本的にこの中でマウスを操作することで，オブジェクトを作成します。この領域でマウスを右クリックすると，十字カーソルがマウスカーソルの位置に移動します。ただし，その図の中で表現できない方向の座標は変化しません。たとえば正面図で右クリックした場合，X軸は右方向，Y軸は下方向ですが，Z方向は表現できませんので(画面奥方向)，十字カーソルはX，Y座標のみに移動します。

また，左クリックは座標確定に使用されますが，この領域内での左クリックによる確定座標はマウスカーソル＋その図内で表現できない方向の十字カーソル座標になります。正面図内ではX，Y座標はマウスカーソルが現在指している座標に，Z座標は十字カーソルのZ座標になるわけです。ややこしいと思うかもしれませんが，実際に使用してみるとそのほうが自然であることがおわかりいただけると思います。

座標はグリッド（青い網目）上の点しか拾うことができません。

・透視図

ポリゴンだから透視図とはいわないのかもしれませんが。要するに作成しているオブジェクトが表示される領域です。カーソルキー左右でY軸回りに，上下でX軸回りに回転します。ただし，適当にぐるぐる回していると，思った方向に回らなくなるこ

とがあります。これはポリゴナイザの内部処理の都合で，決してバグではありません。わけがわからなくなった場合は，UNDOキーを押すことで回転角度を初期化することができます。

大きな物体を作成した場合，表示がこの領域からはみ出ることがあります。バグといえばバグですが，左右が見にくくなる程度で特に支障はありません。

・コマンドエリア

操作モードなどを設定する領域です。それぞれの枠の中をクリックすることで，選択できます。各モードの使用法はのちほど説明します。

・描画属性

左から現在のポリゴン形状，片/両面描画，表面描画色，裏面描画色を表しています。これらの属性はPOINTモード時および，ポリゴン形状と片/両面描画はDELモード時に有効です。また，片面描画では裏面描画色は無効です。

・画面情報

“Center”の下のX,Y,Zの数値がそれぞれ画面上での現在の三面図の中心座標です。その下の“Size”は，現在のグリッドの間隔を表しています。

機能説明

コマンドエリアで設定する各モードの使用法を順に説明します。

・POINT　　　　　バグ出現度★☆☆☆☆

描画属性で示された形状を作成します。描画属性の変更方法については，後述のMODEを参照してください。三面図内でマウスを左クリックすると，その座標からラバーバンドが伸びますので，目的の形状の頂点をポイントしていってください。このとき，描画属性が片面のときは見たい方向から見て右回りに頂点を指定してください（方向を間違えたときは，後述のSCREENで表裏を反転することができます）。

また，頂点を指定中にキャンセルしたい場合は，左右同時クリックするたびに，ひとつ前の頂点をキャンセルしていくことができます。このとき，誤って左ボタンを先に押してしまうと，新たに頂点が確定されてしまいますので，右ボタンを押しながら左クリックするようにするといいでしょう（コマンドボタンを押すことによりすべてをキャンセルすることもできます）。

・CLIP　　　　　バグ出現度★★☆☆☆

すでに作成されたポリゴンの頂点を移動させることができます。移動させたい頂点にカーソルをあわせ，左クリックしてください。その頂点からつながっている辺がラバーバンドで表示されます（表示自体は10本まで）ので，任意座標で左クリックして確定してください。キャンセルはPOINTと同様です。

・DEL　　　　　バグ出現度★★☆☆☆

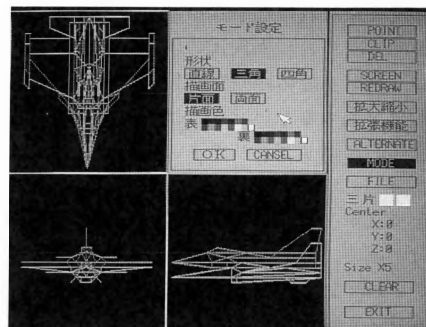
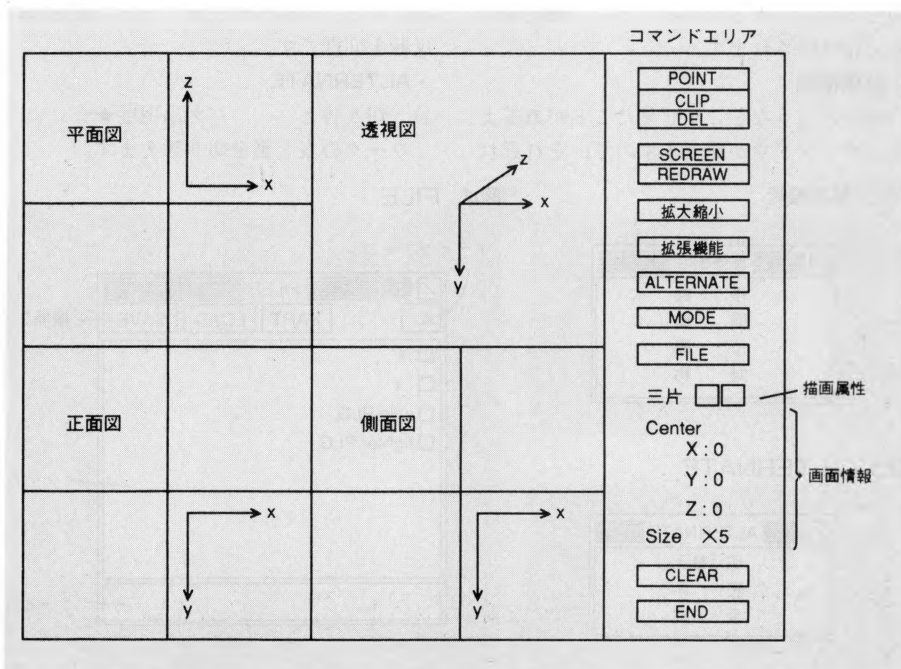
すでに作成されている描画属性で示された形状のポリゴンを削除します。POINTと同じ要領で，削除したいポリゴンを指定してください（右回りであれば頂点指定の順番はポリゴン作成時と同じである必要はありません）。このとき，描画属性が両面描画であれば，両面同時に削除します（どちら回りに頂点を指定しても構いません）。

本来なら頂点の存在しない座標では確定できないなどの方法をとるべきなんでしょうが，手抜きしています。当然のことながら存在しないポリゴンを指定してしまっても，処理が破綻するようなことはありません。

・SCREEN

画面全体が透視図になり，マウスでくるくる回して形状を確認できるようになります。また，作成したポリゴンのうちのひとつが点滅しますが，この点滅しているポリゴン（カレントと呼ぶことにします）に対して以下のような操作を行えます。

1)　カレント移動 バグ出現度★☆☆☆☆

4キーでポリゴンリストの前方，6キーで後方へカレントが移動します。点滅がほかのポリゴンへ移ったことを確認してください。ポリゴンリストの最初で4キー，ま

サブメニューのようす

たは最後で6キーを押したときには，それぞれ最後，最初のポリゴンへカレントが移動します。

2)　カレントポリゴンの移動

バグ出現度★★★☆☆

9キーでポリゴンリストの前方，3キーで後方へ，8キーで最初，2キーで最後へカレントポリゴンが移動します。カレントもポリゴンの移動に伴って移動します。

3)　カレントポリゴンの色変更

バグ出現度★☆☆☆☆

1キーでパレット番号の小さい方向，7キーで大きい方向へ色が変わります。パレットは16色で，標準パレットが割り付けられています。パレット0で1キー，またはパレット15で7キーを押したときには，それぞれパレット15，パレット0になります。

4)　カレントポリゴンの表裏反転

バグ出現度★☆☆☆☆

5キーでポリゴンの表裏（頂点の回り方向）が反転します。

さて，ここでなぜポリゴンリストの順番

図1　画面構成

を変えなければならないかを説明する必要がありますね。横内氏のポリゴナイザは，ポリゴンまでの距離を判定して描画の優先順位を決めるということはやっていません。基本的にポリゴンリストの先頭から順に上書きしていきます。つまりポリゴンの優先順位はポリゴンリスト中で後ろほど優先順位が高く，そのあたりのことはユーザにまかされています（丹氏が自動的に並べ替えるフィルタを作っているようですが)。完全な凸形状であればなんの問題もないのですが，そうでない場合にはいくら巧みに順番を入れ替えても，どこかで妥協しなくてはならないことになると思います（ポリゴンリストで後ろであるほど優先順位が高い点に注意してください)。

なお，カレントポリゴンがほかのポリゴンに隠れて見えない場合，CTRLキーもしくはOPT.1を押しているあいだ，優先表示されます。

・REDRAW

画面を再描画します。後述の裏画面表示のあとに，画面を初期化する場合などに使用してください。

・拡大縮小　　　　　バグ出現度★☆☆☆☆

オブジェクトの拡大縮小ではなく，いわゆるズームイン/アウトです。透視図部分に変なセンスのウィンドウ(?)が開きますので，サイズと中心座標を指定してください。それぞれの枠の中でマウスをクリックすることで値を変えることができ，それに伴って形状が赤いフレームで表示されます（サイズは×1, ×2, ×5, ×10, ×20, ×50の6段階，座標はサイズ単位です)。これらの値は図1の画面情報と同様のもので，確定後はそこに反映されます。

・拡張機能

図2のようなどこかで見たことがあるようなウィンドウが開きますので，それぞれの文字（横に読んでくださいね）にカーソルをあわせてさらにクリックしてください（終了は左上のクローズボックスをクリックしてください)。

1)　移動　　　　　バグ出現度★☆☆☆☆

オブジェクトを平行移動します。透視図部分にウィンドウが現れますので，それぞれの座標の枠にマウスをあわせてクリックしてください。左クリックでサイズ（グリッド間隔）単位で大きく，右クリックで小さくなり，それに伴ってオブジェクトが赤いフレームで表示されます。

RESETをクリックすると，移動量がすべて0になります。移動量を設定し，これでいいというのであればOKを，やめたというのであればCANSELをクリックしてください。

2)　回転　　　　　バグ出現度★☆☆☆☆

オブジェクトを任意軸に対して回転します。それぞれの軸の枠の中で，左クリックにより90度，右クリックで−90度ずつ回転し，それに伴ってオブジェクトが赤いフレームで表示されます。

3)　反転　　　　　バグ出現度★☆☆☆☆

オブジェクトをそれぞれの軸のなす平面に対して反転します。それぞれの平面の枠の中で左クリックすると，枠が反転し，設定されたことを明示します（設定の解除は枠内で右リックします)。平面を2つ設定すると，それらの平面が交差する直線に対して，すべて設定すると原点に対して反転されます。

4)　対称　　　　　バグ出現度★★★★★

オブジェクトからそれぞれの軸のなす平面に対して対称形を生成します。使用法は反転と同様です。

・ALTERNATE

1)　切り替え　　　　　バグ出現度★☆☆☆☆

ワークの表と裏を切り替えます。

2)　表示　　　　　バグ出現度★☆☆☆☆

裏ワークの内容を水色のフレームで表示します。消去するのはREDRAWを使用してください。

3)　合成　　　　　バグ出現度★★★★★

表ワークに裏ワークを合成します。表ワークの後ろに裏ワークを結合しますので，表示の優先順位は裏ワークからの部分のほうが高くなります。

・MODE

透視図部分にウィンドウが開きますので，描画形状，描画面，描画色を設定してください。ただし，描画形状の直線はまだ対応していませんので，選択しないでください。

・FILE

お馴染みの図4のようなウィンドウが表れます。

ドライブアイコン

現在のドライブ名を示しています。ドライブを変更する場合は，このアイコンをクリックしてください。左クリックで正方向，右クリックで逆方向にカレントドライブが移ります。

ファイル表示BOX

カレントディレクトリにあるサブディレクトリおよび拡張子が“.PLG”のファイルが表示されます（ディレクトリには最後に'¥'が付いています)。

それぞれのファイル（ディレクトリ）は先頭にある四角いボタン（のようなもの）をクリックすることで，選択できます。ディレクトリであればカレントがそのディレクトリに移り，ファイルであればファイル名がファイル名入力BOXに転送されたことを確認してください。

また，ファイルがたくさんあってファイル表示BOXに入りきっていないときには，ボタンのある場所以外で左右クリックすることにより,スクロールアップ/ダウンします。

ファイル名入力BOX

このBOXをクリックすることにより，キーボードからファイル名を指定することができます。拡張子は指定しても強制的に“.PLG”になります。

機能アイコン

クリックすることにより，ファイル名入力BOXに表示されているファイルに対して，それぞれの機能が働きます。

1)　PART

バグ出現度★★★★★

図2　拡張機能

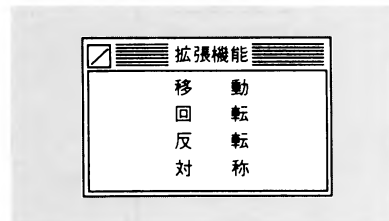



図3　ALTERNATE

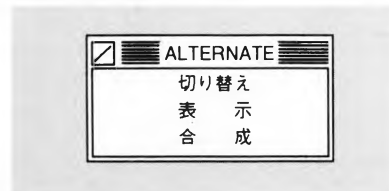
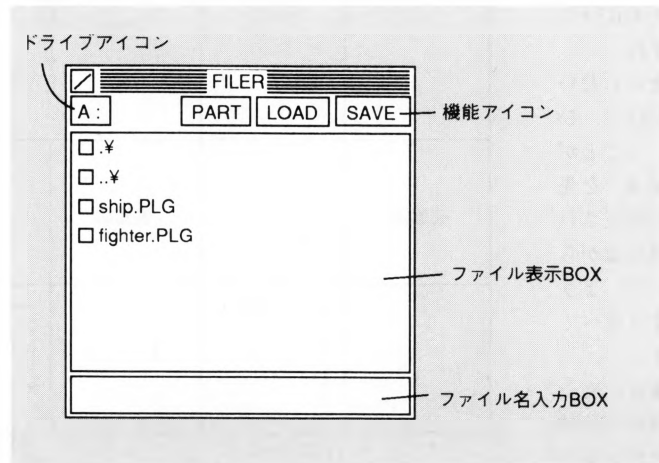



図4　FILE

表ワークとファイルの内容を合成します。表ワークの後ろにファイルを結合しますので，表示の優先順位はファイルからの部分のほうが高くなります。

2)　LOAD　　　　バグ出現度★☆☆☆☆

ファイルから表ワークへオブジェクトをロードします。表ワークの編集中の内容は失われます。

3)　SAVE　　　　バグ出現度★☆☆☆☆

表ワークのオブジェクトをファイルへセーブします。同名のファイルがあった場合，元のファイルの内容は失われます。

・CLEAR

表ワークの内容をクリアします。編集中の内容は失われます。

・EXIT

モデラを終了します。現在編集中の表/裏ワークの内容ともに失われます。

・その他

エスケープキーを押すとなにか現れますが，気にしないでください。あてになりませんから。こんなもの直そうと思えばすぐにできるのですが，優先度の低い部分だったものですから結局直さずじまいでした。

それからROLLUP/DOWNキーでオブジェクトの拡大縮小ができます（表示される大きさはそのままに，グリッド間隔が変わります）。しかし，はっきりいってあまり使ってほしくありません。というのは，拡大縮小によって小数点以下の値は切り捨てられるのですが，たとえば×２から×10まで拡大する場合，×５を経由してしまうために値が狂ってきてしまう場合があるのです。ですから，縮小はやむを得ないとして，拡大はできるだけ後述のPLGCONV.Xでコンバートするときに行ってください。

注意点

最初にもいったとおり，まだかなりのバグがあります(断言できます)。ですからこまめにセーブしてくださいともいいました。しかし，セーブ自体にバグがあったのではお話になりません。では，実際はどうかというと，セーブ自体にはバグはありません（と信じます)。

しかし，ほかの機能で腐ったあとでは，正常に機能するかどうかは保証できません。特に，「ちょっと腐ったけど，元に戻ったから大丈夫だろう」というのは危険です。そういう場合でもいったんクリアするか，または終了することをお勧めします。その際，ファイル名を変えてセーブしておくのも手です。クリアしたあとに正常にロードできるのであれば問題ないはずです。あとはバグ出現度を参考に作業を進めてください。

データ利用法

このモデラでは，オブジェクトのファイルはPLG形式（私が勝手につけただけです）というバイナリのフォーマットを使用しています。そのため，そのままではほかで利用することができません。そこで，PLGを横内氏のポリゴナイザで使用できるようにアセンブラデータに変換するフィルタPLGCONV.Xを用意してあります。

PLGCONV　[/ [X | Y | Z] <整数>]　[ファイル名]

PLGCONVはオブジェクトを整数倍に拡大して変換することができます。２倍に拡大したければ“2”オプションをつけます。また，Ｘ方向のみに拡大したければ“/X2”などとすることができますが，“/XY2”とすることはできません。その場合は“/X2 / Y2”としてください。ファイル名に拡張子はなくても構いませんが，つけても強制的に“.PLG”になり，入力ファイル名のベースネームに拡張子“.S”がついたものが出力ファイル名になります。

また，ワイルドカードにも対応しており，ファイル名を省略するとカレントにあるすべてのPLGファイルが対象になります。ファイル内のラベルなどについては丹氏の記事を参照してください。

反省

本当に申し訳ありません。私としてももう少しどうにかしたかったのですが，時間がなくてどうにもなりませんでした。しかも編集部で報告されたバグがなぜか家で再現できないのです。ひょっとしたら私自身，バグの所在，対策を知っていて，知らず知らずのうちに避けて通っているのかもしれません。とすれば，どこをどうすれば直るのか知っているのでしょうか？　知っていながら教えてくれないなんて，意地悪な私。SLASHのシステム自体が現バージョンからはほど遠いものですし，最新版では現れない症状というのもあるのかもしれません。しかし，このモデラに対しては古いバージョンのほうが都合がいいので，やむなく古いものを使っています。

次の機会にはちゃんと動くようにしときますので，見捨てないで「ああ，こんなもん作ってるんだな，馬鹿な奴」くらいに温かく見守ってやってください。

簡易アトリビュータ　TXED.BAS

SLASHがシェーディング対応になり，ポリゴンのアトリビュート設定が細かくできるようになった。いちいちエディタで記述して変更するのも面倒なので簡単なツールを作ってみた。

SLASHはポリゴンと光源との角度により面の色を決定する。これはテーブルから引いてくるのでそのテーブルを作成する。

光源などからちゃんと計算してやればいいのだが，単純に計算しても光源に向いていない側は単にのっぺりして面白くない。まともにやるにはもっともっと複雑なことが必要だ。ということで，このプログラムでは現実をシミュレートしていない（というよりシミュレートしようとする数式は使っていない)。また，SLASH自体の癖があるので，そのあたりの補正をちゃんとやるのも難しい。よって，私の持っている感覚的なものが最優先されている。

ちなみに「環境光」とか「ハイライト」とかいうパラメータも感覚的なものにすぎないが，結果に応じて経験的に出したものなのでそこそこのことはできると思う。

●メタリックへの道

当面の課題はメタリックな表現である。丹氏から局所的に明度のムラをつければいいという指摘があったが，32階調では不連続な点を出さずにこれを行うのは困難だ。さらに32階調すべてが使えるとはいっても，SLASHで使用する限り明度のダイナミックレンジはさらに狭くなる。

ということで採用されたのが，色相のムラである。もちろんこれは光源の色や環境光の色としても使えなくはないが，本来の目的は別のところにある。

サンプルのALLTESTA.Xで２キーを押すと出てくる金色などがこの機能を使ったものだ。さらにノイズを加えたり，手作業でエディットしたのがSION IVで螺旋状にクルクル回りながらレーザーを出す敵の銀色だ。フラットシェーディングで金属光沢を出そうというのにも無理はあるが，まあこんなものであろう。

●使い方

ツールの使い方だが，適当にマウスでいじってもらえればわかると思う。なお，左右ボタンを同時に押すことで仕様外の値も指定できる。できあがったら「作成」でパレットファイルをアセンブラソースのかたちで作成する。ファイル名はセーブするデータによってユニークな名前となる。

全体の明るさはキーボードから０～９の数字を押すことでシフトされる。それ以外の文字でもなんらかの結果は出る。

作成する色は使用する局面によりまったく変わってくるので背景色を変更できるようにした。バックのメニューにより現在の色を背景として登録することができる。

入力メニューはセーブされたデータのファイル名を与えることで，だいたいの内容を再現する。ファイルのハンドリング関係の処理はX-BASICでは不可能なので手を抜いた。

扱い方は簡単でそこそこの表現力はあるので活用してみてほしい。　　　　　（中野修一）

SLASHのCフロントエンド

_slashlib

Tan Akihiko 丹 明彦

「_SLASH」をC言語でもおいしく使うライブラリ
それが丹氏の制作した「_slashlib」です
まだ試作版ですが，C言語を使う人は愛用してください

横内氏制作のX68000/030用ポリゴナイザライブラリ「SLASH」をC言語から使うためのライブラリを書いた。C言語で宣言したシンボルは，アセンブリ言語のレベルでは"_"(アンダースコア)を先頭につけたものになる。これが「_slashlib」の名前の由来である。

付録ディスクの内容

今回付録ディスクに収録したのは，サンプルプログラムとライブラリファイル，それに開発用パッケージである。

●サンプルプログラム

・alltesta.x(横内氏制作)……アセンブラから利用するSLASHのサンプル。横内氏の記事参照。

・objtest.x……Cから利用する_slashlibのサンプル。いくつかの物体をマウスとキーボードで動かすだけの簡単なサンプル。

●SLASHCOREライブラリファイル(横内氏制作)

アセンブラから利用する。文字どおりSLASHシステムの核である。

・slashlib.a……基本ライブラリ。

・slashRlib.a……ラスター抜き表示を行う設定を行った表示ルーチンを収めたライブラリ。場合によっては表示が数十%高速になる。slashlib.aの中の表示ルーチンにとって代わるものなので，リンク時にはslashlib.aも指定する必要がある。その際，slashRlib.aはslashlib.aより先にリンカに指定する必要がある。

・slashBlib.a……デバッグモードを備えたもの。やはりslashlib.aと同時にリンクし，slashlib.aより先にリンク指定する。

●_slashlibライブラリファイル

SLASHシステムをCから利用するためのフロントエンド関数とちょっと便利なユーティリティ関数を含んだライブラリ。

・_slashlib.a…ライブラリファイル。

・_slashlib.h…ヘッダファイル。構造体の宣言とライブラリ関数のプロトタイプ宣言が収められている。

●開発用パッケージ

開発用パッケージはひとつのディレクトリ構造になっており，この中にSLASH COREも含んでいる。ディレクトリ構成は以下のとおり。以下は，付録ディスクから展開したディレクトリツリーの根元をトップディレクトリとした相対表記である。

・.(トップディレクトリ)
MAKEを実行する場所。サンプルプログラムのソース

・lib¥(_slashlibのソース)
インクルードファイル

・lib¥slashcore(SLASHCOREのソース)

・color¥(色ライブラリ)
アセンブラとCから利用可能

・object¥(物体ライブラリ)
菊地氏のモデラで制作したもの

・font¥(文字オブジェクト)
同じく菊池氏のモデラで制作したもの。Cのサンプルプログラムつき

また，この開発パッケージは今月号から開始した連載「ハードコア3Dエクスタシー」の基本となる。基本的にこのディレクトリ構造のまま使っていただきたい。

サンプルプログラムの使用方法

●起動

コマンドラインからobjtestと打って実行すればよい。コマンドラインオプションはない。スタートアップファイルもない。

●操作

マウスとキーボードを触っていればわかる。というだけではなんだから，簡単に説明する。

・キーを押さずにマウスを移動すると物体が回転する。

・SHIFTキーを押しながらマウスを移動すると光源が移動する。

・CTRLキーを押しながらマウスを移動すると物体の位置が変わる。

・F1キーは物体を赤いクルマに切り替える(デフォルト)。複雑な物体のサンプル。

・F2キーは物体をコアファイターに切り替える。これは菊地氏制作のモデラで作成した形状データを同じく菊地氏制作のコンバータでSLASH用に変換したサンプル。

・F3キーは物体を9棒体に切り替える。これはポリゴンマクロとソートを行うことで複雑な構造の物体を破綻なく表示するサンプル。

・F4キーは物体を戦闘機に切り替える。これもモデラで制作したサンプル。

・F5キーは物体を簡易飛行機に切り替える。これはソートの動作を理解してもらうためのサンプル。

・F6キーは4物体同時表示モード(F1～F4で出てくる物体を一度に表示する)に切り替える。もとの1個表示に戻すにはF1～F5を押せばいい。

・F7キーは物体を構成するポリゴンをソートする。F5キーで出る簡易飛行機を表示している場合以外では使わないこと。また，赤いクルマ(F1)と9棒体(F3)，それに4物体同時表示モード(F6)のときはソートが行われない。

・F10キーは物体を構成するポリゴンの並びを逆にする。ほとんど意味のない機能だが，これをやるとポリゴナイザーが嘘をつきまくる(ポリゴンの前後関係がでたらめになる)のでちょっと変。

●終了

ESCキーを押す。

開発パッケージのコンパイル

●必要な環境

コンパイルのしかたは簡単だ。ただしCプログラムをコンパイルする環境は必要で

ある。特に注意を要するものを次に挙げる。
・MAKE
XCver.2.0に付属するMAKE.Xが必要である。ソフトバンク刊「X680x0 libc」に付属のMAKE.Xではうまくいかなかった。詳しい原因は不明だが，MAKEがチャイルドプロセスとして起動するシェルにCOMMAND.Xを用いているということと，ディレクトリの区切り記号の違いが原因のようだ。ディレクトリの区切り記号はHuman68kは“¥”だがUNIXは“/”で，実にさまざまなトラブルの原因となっている。「X680x0 libc」は，Human68kの上でUNIXライクなコンパイル環境を実現するプロジェクトで（C言語とMAKEの生い立ちを考えれば当然だろう），このトラブルは起こるべくして起こったといえる。
・gcc
Cコンパイラとしてgccを前提としている。私はソフトバンク刊「X68000 Develop.」に収録されたバージョンを用いている。XCではコンパイルしていないが，たぶん通らないだろう。
・Cライブラリ
私は「X680x0 libc」に収録されたCライブラリを用いている。Makefileを修正すれば，XCのライブラリも使用可能なはずである（例によって試していない）。
・アーカイバ
オブジェクトファイル“*.o”をライブラリファイル“*.a”にアーカイブするために必要。「X680x0 libc」に付属のハイスピードアーカイバHAR.Xを用いている。もちろんXCver.1.0に付属の純正アーカイバAR.XやXCver.2.0に付属のライブラリアンLIB.Xを使ってもよい。その場合はMakefileになんらかの変更が必要になるだろう。

●コンパイル方法

1）トップディレクトリに移る。
2）自分の環境に合わせてMakefileを書き換える。最低限確認する必要のあるのが，Cライブラリの位置。付録ディスクに収めたMakefileは，編集室のマシンの設定を反映して，「LIB=c:¥lib」となっているが，多くの人は「LIB=a:¥lib」だろうし，私などは「LIB=a:¥libc¥lib」としている。
3）MAKE dependでライブラリなどを作成する。自動的にサブディレクトリへ移ってMAKE作業を行ってくれるので，基本的にMAKE.Xを実行するのはトップディレクトリだけでよい。
4）MAKEでサンプルプログラムなどを作成する。マシンにもよるが，MAKE dependとMAKEで数分から数十分要する。
5）テンポラリファイルなどを消したい場合は，MAKE cleanを実行すればよい。ライブラリに入れたあとの“*.o”ファイルやテキストエディタのバックアップファイル“*.bak”などを消してくれる。
6）さらに第3者に配布する場合などに，MAKEで生成できるファイルをすべて消去しソースファイルだけを残したい場合は，MAKE distcleanを実行すればよい。サンプルプログラムやライブラリファイルを消去する。MAKE cleanもMAKE distcleanも，サブディレクトリを処理する。

権利関係

SLASHCOREそのものは横内氏の制作であり，その部分の権利関係は氏の記事を参照されたい。

_slashlibの部分の作者は私だが，いたって平凡な作りで，10人のCプログラマがいれば9人まではああいう作りにするのは見えていることもあるし，特に利益を得る目的でのコピーライト主張はしない。このプロジェクトに多少なりとも関わっている，というくらいの認識をしていただければ幸いである。

_slashlibは，試作品としかいえないバージョンで，レジスタの取り回しも思い切り安全サイドに振った効率の悪いものである。少なくとも破壊されるレジスタがわかっているのだから，余分なレジスタ退避をさせないとか，するべきことはたくさんある。さらに効率を追求するのであれば，「X680x0 libc」に収録のIOCSコールやDOSコールのように，ライブラリ関数はすべてインライン展開ということにすべきだろう。また，凶悪なことに，一部のライブラリ関数はテストしていない。とはいえ，サンプルプログラム内で使用した関数は安定して動作しているはずだ。

Cのライブラリレベルでは，構造体を多用している。アセンブリ言語と比べてかなり軟弱な仕様になっていると思う。作者の横内氏の意向を汲むなら，CPUパワーの限界まで攻め込むことこそが美しいのであるから，本来はアセンブラ以外での使用は邪道なのであろう。

もはや，バキバキのアセンブラでプログラムを書けない体になってしまった軟弱なプログラマ（とは私のことだが）に，この_slashlibライブラリを捧げるものである。

今回，一緒に仕事をして横内氏のパワーにはつくづく感服させられた。一線を画したところにいる，というか歴然とした力量の差を感じる。新連載も彼にすべてがかかっているから，「いまの君に病気する権利はない……」などと冗談を飛ばしていたら，逆にこっちが風邪をひいてしまった。まったくもって最悪だ。

SLASH開発パッケージ補足

●そのほかのプログラムについて

容量の関係で付録ディスクに実行ファイルが入ってはいないものの，開発パッケージをコンパイルすると生成されるプログラムがある。
・FONTTEST.X
fontディレクトリに入れた文字オブジェクトのサンプルプログラムである。
起動するとなにも画面に出ないが，かまわずキーを打つと，立体の文字が出てくる。タイプライターみたいなものだ。サポートしている文字は，アルファベットのA～Z（小文字はない）と記号のいくつか（「!」「?」「.」「*」など），それにXシリーズのロゴ（SHIFT＋Xで出る）。マウスで文字を回転させて，CTRLキー併用で光源の方向を変える。ESCキーで終了する。キーリピートもキーバッファもない手抜きなプログラムである。
・FLYTEST.X
フライトシミュレータがどのくらい動くかという興味のもとに，SLASHのパフォーマンスを計る意味で作ったものだ。オブジェクト数や大きさがそれらしいだけで，動きはきわめていいかげん。速度的には望みがありそうなことがわかった。
起動するとランダムな地形の上に浮かぶ飛行機が表示される。マウスで飛行機が左右に傾き，それにつれて旋回する。マウスの左右ボタンで飛行機が前後に動く。回りすぎて止められなくなったときは，F1キーを押して回転を止めることができる。例によってESCキーで終了する。
・PLGTEST.BAT
菊地氏制作のモデラで作った物体を手っとり早く回してみるためのプログラムを作る。
PLGTEST ［拡張子抜きのファイル名］
を実行すると，プログラムを生成する。コンバータPLGCONV.X（同じく菊地氏制作）が必要。実行する前にはバッチファイルをよく読んだほうがいいかもしれない。環境を変えると動かなくなる可能性がある。

●諸注意

これは本文に書いておくべきことかもしれないけれど，SLASHの開発パッケージは，コンパイルしない状態でもハードディスクを2Mバイトほど占有する。コンパイルが完了するまでには，さらにその数倍かかる。開発中はMOが手放せなかった。
付録ディスクには編集室で開発に使っていたそのままの状態（as isってやつだ）でMAKE distcleanしたものを収録してある。
SLASHライブラリをシステムのライブラリ領域にインストールしてしまうのは，待ったほうがいいかもしれない（バージョンアップ進行中）。

SCSIパックンTOWERJACK（その1）

とりあえず1Gバイト

満開製作所 Laido Ryo 頼藤 凌

このところ，毎月のようにOh!Xの一部を占拠しては，なにかと宣伝モードを繰り出す満開製作所だが，Oh!X読者の洗脳を狙って，今度は直接隊員を繰り出すことになった。そういうことで，ひとつよろしく。

SCSIを解放せよ！

ハードディスクの価格低下には目を見張るものがある。240Mバイトが10万円を切り，夢の大容量1Gバイトディスクでさえヘタをすると20万円を切るご時勢である。特にIBM PC/AT互換機のハードディスクは安い。

PCの世界ではIDEというインタフェイスが事実上の標準であって，SCSIはそれほど浸透しているわけではない。バカ安で売られているのもほとんどがIDE対応だ。それでも昨年あたりからようやくSCSI機器も大量かつ低価格に出回り始めた。

MacintoshではSCSIは当たり前。これにPCの市場が加わったこともあり，さらにハードディスク自体の価格崩壊により，いくところにいけば安いSCSI機器があふれかえっている。

X68000のSCSIだが，MacintoshやPC用の機器がほとんど問題なく使える，とはいっても，セクタ長が違うなど，変な物理フォーマットが施されているものは使用できなかったりと，まったく無制限に使えるわけではない。あっちの世界では安いものが大量にあるだけでなく，変なものもちゃんと（？）揃っているのだ。結局はつないでみないとわからない。安くておいしいものにはそれなりにリスクが伴っているのが現状である。

さらに，PCがおいしいのは，タワー型筐体などでは本体内にドライブが簡単に内蔵できるということだろう。PCの場合，主流はタワー型なり，ある程度の大型筐体なので，新しい機器の増設にはドライブを買ってきてケーブルをつなぐだけですんでしまうのだ。

当然のことながら，外付けも安いが，電源や筐体のいらない本体内蔵用のベアドライブであれば，さらに安い。これはMOやCD-ROMなどでも条件は同じである。

巷にこのようなおいしいものがあるのに横目で見ているだけではつまらない。うまいことパックンとしてしまおうというのがTOWERJACKの基本思想である。本来，SCSI機器というものは機種には依存しないはずなのだ。

今回はとりあえず，これらの拠りどころとなるSCSIまわりから見てみよう。

インタフェイスの始まり

昔々のひと昔。時はまだ混沌の時代。ハードディスク（以下HDと略）とコンピュータをつなぐ方法なぞ乱立だらけで，HDそのものもまだトライアンドエラーの頃のこと。アメリカに，シュガート・アソシエイツという会社があったそうな。この会社，Hな，もといHDを作るのがとっても得意で，毎日毎日せっせせっせと精出してHDを作っておったとな。（中略）

なんにつけてもそうなんだが，時はいろんなものを解決する。そうこうしているうちに，いつの間にやらシュガートがもっとも影響力を持つHDメーカーとなり，ほかのメーカーは，シュガートのインタフェイスに倣うようになっていった。（後略）って，これじゃなんだかよくわからないか。

なにごとにつけても大きな木を拠りどころにしたほうが物事は楽に運ぶもので，最近の表計算ソフトは1-2-3のデータが使えるようになっているのが当たり前，ワープロは「一太郎の文書が読み込めます！」というのが売りで，Human68kに至っては，「MS-DOSのディスクも読めます」とくる。改めて考えてみればすごいことだけど，……いや，本当にありがたいことだ。

ともかく，HDのほうは既成事実的な経緯で業界標準のShugart Associates Standard Interface（Standardじゃなくて Systemだという見解もある），略してSASIというインタフェイスができあがったのだ。これもひとえに，楽して儲ける人間のサガのお陰さま。いや，本当にありがたい。

SCSIの夜明け

ハードディスク用のインタフェイスとはいえ，SASIは装置の論理アドレスに書き込むとか，ホストは大雑把なコマンドを送るだけで実行は装置側でという汎用性の高い方法を用いていたので，どうせならもっと幅広い機器に対応させようという動きが出てきた。当時はやり始めたLANでHDを共有するなんて考えは，出るのが当然。いっそのことコンピュータ同士もつながったほうがよいと考える者も出てくるに至って，ついにアメリカのANSIという団体（日本でいうJISみたいなところ）が，SASIをもとにさらに拡張性のあるインタフェイスを作ることを試み始めた。時は80年代半ば。X68000がいままさに生まれんとする頃，Small Computer System Interface，通称SCSIの策定が始まった。

名称のつけ方ってのがなかなか粋で，洒落てやがるぜアメリカン。江戸っ子じゃないけどANSI屋の生まれか。これから作ろうという規格に1メーカーの名称を使うことなど当然ありえないので，SASIとSCSI，ゴロは似てても心は別物。正式名称はかな

り違う。

SCSIはその名のとおり，もともとは“小型コンピュータ”，はっきりいえばパソコンレベルで使用することを目的としたインタフェイスだった。最近ではワークステーションなどにも採用されてしまって，不満をたらたらいわれていたりするが，本末転倒というものだろう。また，ここ数年のパソコンの飛躍的な進歩のお陰で，少々古めかしいインタフェイスと変わりつつあるが，それも努力次第でなんとかなるって。パソコンってのは，もともとユーザーの努力なくして成り立つものではないからね。

さて，SCSIがまず最初に世に出たのは1986年。とりあえず，この段階で同じインタフェイス上に，ハードディスク，磁気テープ，プリンタ，プロセッサ，追記型ディスクと，CD-ROMというたぐいのデバイスを8つまで接続することが可能となった。実際にはパソコン（が利用しているインタフェイス部分）がひとつのプロセッサにあたるので，周辺機器としては7つまでではあるのだが。

パソコンとSCSI周辺機器との接続は両端にターミネータ（終端抵抗）というものをつけた1筋のケーブルに機器をぶら下げるかたちにする。ちょうどパン食い競争のパンをぶら下げている図を思い浮かべてもらえばわかりやすい。両端の柱がターミネータ。パンがSCSI機器にあたる。

一般にHDには，コネクタが2つついている。たいてい無表示だが，ときたま片方からIN，もう片方からOUTするようになっているものを見かける。しかしHDの中では単に線が2つに分岐しているだけのはずなので，特に作りが特殊でもない限り，IN・OUTを逆につないでも，なんの問題もないと思う。そもそも，SCSIでは厳密にはパソコン（の持つSCSI部分）もひとつの周辺機器の扱いなので，INやOUTの方向はいつも同じ向きとは限らないのだ。

ところで，ターミネータというのは，単なるチープな集合抵抗である。しかし，こいつがSCSIではかなり重要な役目を果たしていて，ひとつにはSCSI機器から出た信号を，接続ケーブルの端々までもれなく伝えるため。またもうひとつには端まで伝わった信号を吸収するために，必ず接続しておく必要がある。「ターミネータをつけなくても動作をする」というレポートをあちこちで見かけるが，これは，たまたまうまく動いているにすぎない。ターミネータなしでは，SCSI周辺機器にノイズ混じりのデータを送っているのと同じことになるからだ。ケーブルの端まで伝わった信号は，ケーブルの末端で反射を起こしノイズと化してしまう。そう高いものでもないのでなるべくケーブルの端にはターミネータを接続するようにしたほうが身のためだろう（とはいえ，たまにターミネータを外さないとまったく動かない場合もあるので注意）。

逆に間にターミネータを入れてしまった場合は，そこで信号が吸収され，端まで伝わらなくなってしまうので，これまたうまく動作しなくなる。場合によっては過剰電流が流れて機器を破壊するおそれもある。マニュアルどおりにやればなんの問題もないことなんだけれどね。

……のハズだったのに

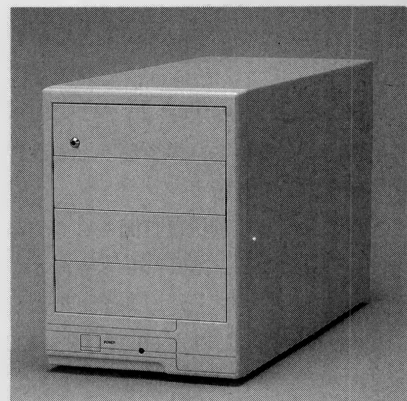
しかし1986年，当時のSCSIは，まだ規格をメーカー独自で解釈できる部分が残っており，多くの“方言”を許していた。また細かな取り決めはしてあっても，任意でサポートしなくてもよい部分も多く，“SCSI対応”とはいっても，各メーカーの独自の“SCSI”ボードでなければパソコンと接続できない周辺機器が数多くあったようだ。このため，当時発売された“SCSI”ボードや周辺機器は，お互いが自社の製品であるかどうかをチェックして，それ以外では起動しないようにデキているものが多くある。たとえば噂のPC-9801……。

昔のX68000と同じく，PC-9801のSCSIも拡張スロットにボードであとから付け足すかたちになっていた。PC9801-55という

あ，すいません

いきなりだが，9月半ばから満開製作所ではその1Gバイトハードディスクを発売しているはずである。名称は「満開式拡張型硬盤駆動装置壱號：MK-HD1-EX」。別名，「TOWERJACK 1ギガHDセット」である。価格はケーブル同梱で開発記念特価25万円（税込み）の予定。

これぞ“満開式”とでもいわんばかりの，無骨な筐体「TOWERJACK」には，手始めに1Gバイトのみが内蔵されている。将来的には満開製作所指定のSCSI機器（MOやバックアップストリーマなど）が4台まで内蔵できる作りになっており，たとえばもう1Gバイト増設するためのHDも20万円前後で販売される予定である。ハードディスクも1Mバイトあたり約200円。フロッピー並みの価格になってきたというわけだ。

製品の性格を考えるとわかるように，別にX68000専用の機械というわけではない。とりあえず，まともなSCSIがあればどんな機種でも使用できるはずだ。X68000とFM TOWNSでは動作確認されているし，ケーブルさえなんとかすれば，Macintosh，もちろんPC/AT互換機でも使用できるだろう。SCSIさえあればPC-9801以外のあらゆる機種に接続できるはずである。

正式なことはまた広告も出るでしょうが，Oh!Xのカラーページで宍戸留美ちゃんを見つけてお問い合わせください。

では，ばははーい。

SCSIボードを利用したPC-9801用のSCSIを，とりあえず，親しみを込めて“けっ，98-SCSI”と呼んでやることにしよう。

この“けっ，98-SCSI”は，すでに衆知のとおり，PC-9801用以外の（普通の）SCSI周辺機器は，全滅といっていいほどつながらない。理由はしごく単純で，“けっ，98-SCSI”ボードが「周辺機器がNECの製品かどうかをチェックし，そうでなければ起動すらしない」ようにできているからだ。

さて，なぜNECがこのようなボードを世に出さねばならなかったかというと，「SCSI周辺機器とやりとりするパラメータを間違えて独自解釈しているおそれがあったから」なのだろうと推測される。実際，どうやら他社の解釈とは相違のある部分が出ていて，仮にハード的には接続できても，HDなどにデタラメなデータを書き込むおそれすらあるのだそうな。そうだとすると，すんなりつながってデータを破壊されるより，このNECチェックは，ある種の親切だと考えられなくもない。

時は流れて，はや6年。X680x0もいまではSCSIを内蔵するようになっているし，PC-9801は新しいタイプの（98？）SCSIボードを出すようである。多少なりとも改善されればよいのだけどね。

こうして，徐々に状況は打開されつつある。いつの日にかPC-9801にもちゃんとしたSCSIが内蔵される日がこよう。なにしろ，私が最初に買ったPC-6001にはRS-232Cインタフェイスすらついてなかったのだ（ただ，いまだにPC-9801の“けっ，RS-232C”はFAXモデムをつなぐときなんかに問題の出るものがあるらしいんスけど）。

標準のSCSIへ

PC-9801はさておき，発表当初こそ接続に難のあったSCSIも，メーカーが中心になって規格の詳細を練り直したCCS（Common Command Set）というものが発表されてからはずいぶんと問題が少なくなった。このCCSは主にソフト側の規格で，実際にはSCSI-1.5とも呼べるものである。

このCCSでやっと機器同士の相性問題にケリをつけたSCSIは，現在速度面でより改良を加えられたSCSI-2になって今年から出まわりつつある。この調子なら2100年にはSCSI-20なんてのが出てくる……，なんてことがあるかどうかはわからないが，SCSIに対する要望はそれだけ強く，すでに大量に使われているのだ。

ちなみに，MOやスキャナが正式にSCSIにつながることが決められたのはSCSI-2からで，X68000でもSCSIボードに載っているBIOS-ROMなどは古いSCSIにしか対応していない。このためSCSIボードではMOなどからは起動できないようだが，ほかのデバイスから起動をすれば，お馴染みのINQPATCH.X（毎度お騒がせしてますが，電脳倶楽部というディスクマガジンに掲載された）でソフト的にドライブを認識させることは可能になる。

どこかの奇特な会社がROMの交換サービスなんてのを始めれば，起動もできるようになるんスけどねー，なんて拝んでみるといいかもしれない（笑）。

ともかく，皆さんのX68000にSCSIがついてなければ，スロットをつぶすことになってもSCSIボードだけは買っておいたほうがいい。

Human68kなどでもver.3にもなると，SCSI-2を考慮していて，なんの設定がなくとも，最近のHDやMOがドライブとして認識されるようになっている。逆にいまだに一般性に欠けるCD-ROMは，CD-ROMのドライブごとに対応したデバイスドライバが必要らしい。これは，最前線のAT互換機の世界でも同じだからしょうがない。

ひと夏の体験

とまぁ，ここまではis はっきりいえば受け売りで，聞きかじりと寄せ集めでごたごた書いてきたわけなのだけれど，ここからは，実際に私が体験したことをお話ししよう。

唐突だが，パソコン用HDにG（ギガ。海の向こうでは，かっこよく“ギグ”とも発音するらしい）の桁が現れ始めたのは去年の夏のことだった。

それまで巨大なワークステーションと，ごく一部のメーカーの実験室だけで丁重に扱われていた超大容量HD。昨年頃からばりばりと広まり始めたMS-DOSのver.5とOS-2が引き金で，まずはAT互換機の世界からお目見えし始めた。ま，それまでのMS-DOSのバージョンでは1Gのハードディスクは扱えなかったらしいのだけれど，ちょうどHDを大量に食いつぶすWindowsなんかの進展と合致してこれが世間を席捲し始めたわけなのだ。

日本のAT互換機ブームとか，PC-9801でIDE採用とかいう関係もあると思うが，このところ海外のHDメーカー（Quantumとかんかメーカー（Maxtorなんてところ）は勢いづいている。日本のメーカーがPC-9801用で日々安泰に送っている頃，実は海外のメーカーは30Mバイトを足掛かりに60・120・240と倍々式に容量を増やしていた（倍にするなら開発コストは比較的安くてすむらしい）。それを一気に，3倍ごとに増やし始め，1Gバイトを超えたのがちょうど去年の夏あたり。最初に出したのは，富士通だったか，東芝だったかの日本企業だと思うが，海外メーカーがいま一気に低価格化を武器に巻き返しをはかりつつある，というのが現状である。

PC-9801でいまだに現役で使われているver.3.3xというMS-DOSでは，どんなに大容量のHDを手に入れても128Mバイトごとに分割しなきゃならないらしいが，こういう世界に固執していると，日本はまるごと心中ということになりかねない。

ただ，隣の芝生が枯れていくのを見て，ニヤリとばかりもしてられない。我らがHuman68kも，DOSコールのDSKFRE($FF36)なんてのは2Gバイトまでしか対応していないフシがある。ま，これぐらいなら仮に4GバイトのHDなんかを手に入れたとしても，2Gバイト2つに分ければ済む話といやぁ，それまでだけど。

さておき，ひょんなところから，X680x0でも使える1GバイトのHDを作成したので，それについてつらつらと書いてみることにしよう。Human68k ver.3で使うなら，SCSIのコネクタで接続するだけでなんの問題もなく動作するし，ver.2でもSCSIDRVを登録すれば簡単に扱える。もちろんシステムを転送すれば当然起動もできる。

Human68kでは16ビットFATが採用されているので，最大クラスタ数は65536個となる。ふつうは1クラスタ1024バイトなので，64Mバイトまではそのままの状態で扱える。

これが64Mバイトを超えるとクラスタは倍々で拡大されていくことになる。たとえ

ば，3.5インチMOなんかは121Mバイトだから，1クラスタ2048バイト，5インチMOだと1クラスタ8Kバイトとなる（1パーティションで区切った場合)。そして，1Gバイトを超えるHDを1パーティションで区切ると，1クラスタは32Kバイトにも達する。

これはなにを意味しているかというと，ファイルの最小ディスク占有量がどれだけになるかということである。

CONFIG.SYSを書けばディスクの残り容量が32Kバイト減り，CONFIG.BAKとかAUTOEXEC.BAT(.BAK)などがあると，それだけで128Kバイトが消費されるということになる。まあ，実際にはあまり大きなパーティションを切るよりも多少は分割したほうが使い勝手がよくなるので，こんなことをする人はいないかもしれないが。

さて，1クラスタの最大サイズは64Kバイトまでとなっているのでヒuman68kで扱える1パーティションの最大サイズは4Gバイトまでとなる。先ほどのDSKFREなどの問題でディスクの空き容量表示はおかしくなるかもしれないが，扱えることは扱える(はずだが試したわけではない)。ドライブ数の制限などもあわせると，理論上，Human68kで管理できるファイルの最大量は96Gバイトということになる（実際には+2.4Mバイトだが，誤差の範囲だろう)。内蔵HDの1000倍以上というほとんど絵空ごとの世界のようにも思えるが，SCSIの勢いを見れば，そのうちこれでも足りなくなることがあるのかもしれない。

贅沢な悩み？

で，とりあえず1Gバイトである。問題は，いかにしてこの巨大な容量を食いつぶすかだ。まずは，計算してみる。

1GバイトのHDにAD PCMを記録したとしよう。X680x0のAD PCMは1秒に最大7,800バイトを消費する。当然のことながら，1Gバイトは1024Mバイト。1Mバイトは1024Kバイトで，1Kバイトは1024バイトだから，1Gバイトは1億バイト。つまり，1GバイトのHDには1億÷7800……。えーいめんどうだが，ともかく約38時間の録音が可能なのだ。

丸1日半，PCMを鳴らしっぱなし。うけけ，こんな贅沢があろうか。……と思って，録音を始めたが，8時間ほどでやっぱり飽きた。持ち歌のレパートリーが切れたというのもあるが（ウソ)，据膳のあるときには，意外と腹は減らないものだな。とりあえず，200Mバイトを浪費したけど。

次に画像を記録してみると仮定する。手前みそだが，現在までに電脳倶楽部に掲載された画像ファイルは優に700を超え，PICなどを使っているとはいえ，容量も10M近くになっている。仮にこれをベタでセーブし直したとしよう。しかしながら1Gのハードディスクを埋め尽くすことなどできはしない。とりあえず，350Mバイトを浪費するけど，それでもやっと半分か……。

ついでだが，最近話題のアウトラインフォントを移してみたと仮定する。SXのフォントマネージャでツァイトのJGフォントが利用できるというのは，もう皆さんご存じのとおりだろう。私も勢い余って買ったクチで，とりあえず速度さえなんとかなればと思ってたりするが，これが3書体で7Mバイトを使用する。

恐るべきことに，JGのフォントはWindows用のフォントなどと比較してもわりと軽めのほうで，この夏キヤノン製のWindows用のフォントドライバのバージョンアップサービスを受けるときに，お遊びでペン字体をつけてみたら，1書体だけでHDを4Mバイト近く必要とした。マルチフォントのワープロが出てくれば，アウトラインフォントは，次から次へほしくなるもので，私が使っているAT互換機のHDはすでに25Mバイトを超えるフォントが収納されている。

他人ごとじゃないぞ。SXはフォントドライバさえ作成してしまえば，TrueTypeだって，キヤノン系のフォントにだって対応するに決まってるのだから。これで，自分でもフォントを作り始めちゃったら，それこそもう100Mバイト程度は軽くいる。

さて，最後に，動画を記録した場合について考えてみよう。現在X680x0で標準となっているフルカラーの動画規格ははっきりいえば，まだない。しかし，SXの動画ウィンドウを見ていても感じられるように，遠からぬ将来，X680x0でも自然動画を扱える時代がやってくるに違いない。そんな希望的観測を抱きながらも，とりあえず，いまのところは先陣を切って茨の道を歩みつつあるWindows関連のデータを参考に推測をしてみるとする。

パソコンレベルとなると，いまの段階ではアメリカにおける動画の規格もまるで統一されていない。Video for Windowsなんてのが巧いことやってくれるんだろうが，いまのところ，動画圧縮規格はかなり乱立ぎみである。プリンタと同じでデバイスドライバ次第なのかもしれないが，ま，ある程度落ち着いたところで，パックリやったほうがウハウハだと，私もそう思う。

さておき，現在あるWindowsの動画関連のメーカー側の発表を鵜呑みにすれば，320×240ドット，フルカラー（24ビット色)，毎秒30コマ程度のテレビ画像並み。これで要するディスクスペースが，1分間で9M〜50Mバイトだといわれている。1GバイトのHDでも，これでは20分，よくても2時間程度しか録画できないのだ。

デスクトップムービーを作ろうと思った場合，動画を編集するディスク容量などを考えると，20分程度の短編映画を1本作るのがやっとというところではなかろうか。こう考えると，1Gバイトといえども，そう広くは感じられなくなってしまうもんだ。

Gの時代へ

そうじゃなくても，最近ではゲームもディスクの枚数が増え，HDにインストールできるものが多くなってきた。MOなどでデータのやり取りをする集団なんてのも増えたし，それモンの写真集がCDで出始める時代だ。しかし，いくらかずつ改善されているとはいえ，FDにしてもCDにしてもMOにしても，総じてまだまだ遅いのだ。

シリコンディスクは材料の1工場が操業できなくなるだけで手の届かない価格に跳ね上がるし，フラッシュメモリも一向に使えるところまで降りてこない。結局，HDに取って代わるメディアは，まだしばらくは現れてこないんじゃないだろうか。みんなとっかえひっかえやりくりし，HDにデータを移しては使い続けるしかない。それが今後10年は続くと思う。

いつまでも数十，数百MぽっちのHDなんかでやっちゃ いけませんって。そろそろ皆さんも心したほうがいいスよ，ほんと。

Mは終わった。そしていま，Gが始まる。

Oh!X LIVE in '93

X68000・Z-MUSIC用
(SC-55対応)

未来予想図Ⅱ

Yamada Kai 山田　開

X68000・Z-MUSIC
+PCM8用

©SEGA
「OutRun」より

PASSING BREEZE

Tateno Nobu 館野　暢

秋はお出かけの季節。美しい風景のなかをすてきな誰かとかっとばすのもいいかも。お留守番のX68000にはこの2曲を打ち込んで，秋の夜長を楽しんでくださいね。ということで，出かけてしまった進藤氏のコラムはお休みです。

ブレーキ5回，踏んでますか？

さて，今月の１曲目はDreams Come Trueの古い（?）ナンバーをお届けしましょう。タイトルは「未来予想図II」です。この曲はアルバム「LOVE GOES ON…」に収録されていた名曲です。「～II」といっても，ダッシュとかターボはありませんのであしからず。アルバム自体はもう４年近く前のものになるのでしょうか。このLIVE inのページでも，同アルバムより「うれしい！たのしい！大好き！」が掲載されたことがあります。

この作品の演奏にはSC-55同等品が必要です。内蔵音源も使用しているので，ミキシングにも注意してくださいね。

さて，注目すべきポイントは，ボーカルの吉田美和の声を何で表現したかという点にあります。この作品ではフルートを使ったようですね。インストルメンタル風に聴こえてしまうという点があるものの，それだけマッチしているともいえます。曲調もあわせて考えると，ひとまず成功してるというべきでしょう。泥臭いボーカルにはなっていないので，まだ研究の余地がありますね。

演奏する前に，サンプリングファイルを加工する必要があります。なにはともあれ，ZVT.Xを起動してください。まずsnare2.pcmにchorusをかけ，volume50%にてSD.PCMというファイル名でセーブします。同様に，fck.pcmをchorus, volume30%でBD.PCMとしてセーブしてください。手元にエフェクタ持ってるぜっという人なら，.CNFファイルのvオプションだけ変更してみるのもいいでしょう。

LOVE GOES ON…

波風うけとめて

２曲目にお届けするナンバーはセガの名作体感ゲーム「OutRun」より，「PASSING BREEZE」です。もちろん，みなさんご存じですよね？　ゲームセンターに行けば，いまでも置いてあるところもあるでしょう。オープンのフェラーリテスタロッサを駆って５つのゴールを目指すドライブゲームです。

OutRun

ゲームミュージックとしては，年代を考えれば十分に古いと呼ばれるようなモノですが，いま聴いても新鮮に思えるのはそれだけ曲が素晴らしいということなのでしょうか。発表当時には，フュージョン系を意識して作曲した，とコメントされていたと思います。

この作品の演奏にはPCM8.Xが必要です。つ・ま・り，内蔵音源だけで演奏可能なのです。MIDIの作品が多いなか，健闘しているといえるでしょう。もちろん，曲のデキも折り紙つきです。10MHzマシンでもちゃんと演奏されるので，安心して入力してくださいね。

作者の館野君によると，原曲に忠実という点にこだわったため，エフェクタがかかっているCD版とはちょっと違うとか。確かに，CD版と聴き比べるとエフェクタが欲しくなります。できれば，エキサイターとリバーブをほんの隠し味程度にかけるとよろしいでしょう。十分にCDと対抗できるようになります。当たり前のようですが，エフェクタを通さずに聴いてもOKです！

この曲はいろいろな人の意見を参考に作ったようですね。自分のまわりの実力者に聴いてもらうのも，上達への早道ですよ。

(SIVA)

リスト１　未来予想図Ⅱ

```
 1: /--------------------------------------------------/
 2: /           L O V E  G O S E  O N …               /
 3: /                                                  /
 4: /                 未来予想図II                     /
 5: /                                                  /
 6: /            (C) Dreams Come True                  /
 7: /                                                  /
 8: /            Copy By. Kai  Yamada                  /
 9: /--------------------------------------------------/
10:
11: .ADPCM_BLOCK_DATA = MI.ZPD
12:
13: (v1,0,
14: /       AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
15:         61, 15,  3,  1,200,127,  0,  0,  0,  3,  0
16: /       AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
17:          4,  5,  5,  2,  3, 24,  0,  2,  0,  0,  0
18:          4,  5,  5,  2,  3, 13,  0,  1,  0,  0,  0
19:          4,  5,  5,  2,  3,  6,  0,  1,  0,  0,  0
20:          4,  5,  5,  2,  3,  5,  0,  1,  0,  0,  0)
```

```
     21:
     22: (v2,0,
     23: /          AF   OM   WF   SY   SP PMD AMD PMS AMS PAN
     24:            56, 15,  3,  1,200,127,  0,  0,  0,  3,  0
     25: /          AR   DR   SR   RR   SL   OL   KS   ML DT1 DT2 AME
     26:            16, 13,  6,  3,  3, 26,  0,  3,  6,  0,  0
     27:            19,  2,  6,  5,  0, 25,  0,  1,  3,  0,  0
     28:            16,  2,  6,  3,  0, 26,  0,  1,  3,  0,  0
     29:            17,  2,  6, 11,  0,  0,  0,  1,  3,  0,  0)
     30:
     31: (v3,0,
     32: /          AF   OM   WF   SY   SP PMD AMD PMS AMS PAN
     33:             2, 15,  3,  1,200,127,  0,  3,  0,  3,  0
     34: /          AR   DR   SR   RR   SL   OL   KS   ML DT1 DT2 AME
     35:            31,  8,  8,  6,  2, 35,  0,  3,  7,  0,  0
     36:            31, 10, 10,  6,  3, 26,  0,  2,  0,  0,  0
     37:            31, 10, 10,  6,  2, 31,  0,  3,  7,  0,  0
     38:            31, 10, 10,  6,  3,  0,  0,  4,  0,  0,  0)
     39:
     40: (i)
     41: (b1)
     42: (d1)
     43: (m1,8000)(a1,1)
     44: (m2,8000)(a2,2)
     45: (m3,8000)(a3,3)
     46: (m4,8000)(a4,4)
     47: (m5,8000)(a5,5)
     48: (m6,8000)(a6,6)
     49: (m7,8000)(a7,7)
     50: (m8,8000)(a7,8)
     51: (m15,8000)(a1,15)
     52: (m10,8000)(a10,10)
     53: (m11,8000)(a10,11)
     54: (m17,8000)(a17,17)
     55: (m18,8000)(a18,18)
     56: (m25,8000)(a25,25)
     57:
     58: /-----------------------------------------------
     59:
     60: .roland_exclusive $10,$42={$40,$00,$7f,$00}
     61: .sc55_v_reserve $10={6,5,2,2,2,2,2,0,0,3,0,0,0,0,0,0}
     62: .sc55_reverb $10={5,2,0,72,50,43,0}
     63:
     64: /-----------------------------------------------
     65:
     66: (t1)     @i$41,$10,$42
     67: (t1)       i0  @1  o3  v8  l16  q8  p3  @e0,0  t75
     68: (t1)     @y$01,$64,$33r4g-ff-e-'e--2<e--g-''f-2<f-a-'<'g-
8.b-''b-8<d-''a-<c-''g-4.b-'
     69: (t1)     @e100,10@y$01,$64,$35 g-ff-e-v13@u80'e--2g-b--'
     70: (t1)     v14'f-2a-<c-'v15'g-8.b-''b-8<d-''a-<c-''g-b-'l20
     71: (t1)     @u100l4a-.b-2&b-8a-.b-2<d-16d-16&d-1&d-2r2
     72: (t1)     o4v15@u50|:4'g-a-b-<d-'@u+5:|@u50|:6'a-<c-e-'@u+
5:|
     73: (t1)     @u70'a-<c-e-g-'@u60'a-<c-e-f'|:4'g-a-b-<d-'@u+3:
|
     74: (t1)     |:4'g-<c-d-e-':||:4'fa-b-<d-':||:4'e-g-a-<c-':|r
<
     75: (t1)     @u90'g-8b-''fa-''e-g-''d-8f'>|:4'g-a-b-<d-':|
     76: (t1)     |:7'a-<c-e-':|'a-<c-e-f'|:4'g-a-b-<d-':|
     77: (t1)     |:4'g-<c-d-e-':||:4'fa-b-<d-':|@u91|:'e-g-a-<c-'
:|
     78: (t1)     @u92|:'d-g-a-<c-':|@u93|:3'g-a-b-<d-':|@u95'ce-g
-b-'
     79: (t1)     v16@u100|:4'c-e-g-b-':||:4'd-e-fa-':||:'c-e-g-b-
':||:'d-e-fa-':|>|:'b-<d-fa-':|
     80: (t1)     |:'b-<d-f-g':||:'a-<c-e-g-':||:'a-<c-d-f':||:'b-
<d-fa-':||:'b-<d-e-g-':|
     81: (t1)     <|:'c-e-g-b-':||:'ce-g-b-':|q5|:4'c-8e-g-b-':|'c
-8e-g-b-'r.
     82: (t1)     q8@u110|:4'g-a-b-<d-':||:'fa-<c-e-':||:'fa-b-<e-
-':||:'e-g-b-<d-':|
     83: (t1)     |:'d-g-b-<d-':||:'c-e-g-b-':||:'d-fa-<c-':|
     84: (t1)     |:4'g-a-b-<d-':||:'fa-<c-e-':||:'fa-b-<e--':||:'
e-g-b-<d-':|
     85: (t1)     |:'e--g-b-<d-':||:'d-g-b-<d-':||:'cg-b-':|
     86: (t1)     @y$01,$64,$60@u70|:'c-e-g-b-':||:'c-e-fa-':|>|:'
b-<d-fa-':|
     87: (t1)     |:'b-<d-f-g':||:'a-<c-e-g-':|<|:'c-d-e-g-':||:'>
b-<d-fa-':|
     88: (t1)     'e-g<c-d-''e-gb-<d-'|:4'c-e-g-b-':|>|:'b-<d-fa-'
:|
     89: (t1)     |:'b-<d-e-g-':||:'a-<c-e-g-':|<|:'c-d-e-g-':|
     90: (t1)     |:|:'e--g-b--':||:'f-a-<c-':||:4'g-a-b-<d-':|:|
     91: (t1)     o4v15@u90|:4'g-a-b-<d-':||:6'a-<c-e-':|
     92: (t1)     '<e-g-''<e-f'|:4'g-a-b-<d-':|@u90|:'g-<c-d-e-':|
     93: (t1)     l8@u80'g-<c-d-e-'@u95g-<e->g-l4@u90|:'fa-b-<d-':
|
     94: (t1)     l8@u80'fa-b-<d-'@u95r<d->fl4@u90|:4'e-g-a-<c-':|
r<
     95: (t1)     @u100'g-8b-''fa-''e-g-''d-8f'>|:4'g-a-b-<d-':|
     96: (t1)     |:7'a-<c-e-':|'a-<c-e-f'|:4'g-a-b-<d-':|
     97: (t1)     |:4'g-<c-d-e-':||:4'fa-b-<d-':|@u101|:'e-g-a-<c-
':|
     98: (t1)     @u102|:'d-g-a-<c-':|@u103|:3'g-a-b-<d-':|@u105'c
e-g-b-'
     99: (t1)     v16@u100|:4'c-e-g-b-':||:4'd-e-fa-':||:'c-e-g-b-
':||:'d-e-fa-':|>|:'b-<d-fa-':|
    100: (t1)     |:'b-<d-f-g':||:'a-<c-e-g-':||:'a-<c-d-f':||:'b-
<d-fa-':||:'b-<d-e-g-':|
    101: (t1)     <|:'c-e-g-b-':||:'ce-g-b-':|q5|:4'c-8e-g-b-':|'c
-8e-g-b-'r.
    102: (t1)     q8@u115|:4'g-a-b-<d-':||:'fa-<c-e-':||:'fa-b-<e-
-':||:'e-g-b-<d-':|
    103: (t1)     |:'d-g-b-<d-':||:'c-e-g-b-':||:'d-fa-<c-':|
    104: (t1)     |:4'g-a-b-<d-':||:'fa-<c-e-':||:'fa-b-<e--':||:'
e-g-b-<d-':|
    105: (t1)     |:'e--g-b-<d-':||:'d-g-b-<d-':||:'cg-b-':|
    106: (t1)     @u90|:'c-e-g-b-':||:'c-e-fa-':|>|:'b-<d-fa-':|
    107: (t1)     |:'b-<d-f-g':||:'a-<c-e-g-':|<|:'c-d-e-g-':||:'>
b-<d-fa-':|
    108: (t1)     'e-g<c-d-''e-gb-<d-'|:4'c-e-g-b-':|>|:'b-<d-fa-'
:|
    109: (t1)     |:'b-<d-e-g-':||:'a-<c-e-g-':|<|:'c-d-e-g-':|
    110: (t1)     |:4'g-a-b-<d-':||:4'c-e-g-b-':|>|:'b-<d-fa-':||:
'b-<d-e-g-':|
    111: (t1)     |:'a-<c-e-g-'~:|<|:'c-d-e-g-':||:4'c-e-g-b-':|
    112: (t1)     >|:4'a-<c-e-g-':|<|:'e--g-b--':|'c-e-g-a-'r
    113: (t1)     v11@p58@u127'e--g-b--'r8v12e--8v11|:'f-a-<c-'_8:
||:3'g-a-b-<d-'_5:|
    114: (t1)     v11'g-a-b-<d-'v11|:'e--g-b--'_5:|v11'f-a-<c-'v12
f-v11|:3'g-a-b-<d-'_8:|
    115: (t1)     v11'g-a-b-<d-'v10g-8v11e--8v13'b--<e--g-'v11_5
    116: (t1)     'f-a-<c-'~2'c-f-a-'|:3'g-a-b-<d-'_8:|v11'g-a-b-<
d-'
    117: (t1)     l8v10g-d'b--<e--g-'a'f-4a-<c-'a-el8v8|:8'g-b-<d-
'~8:|
    118: (t1)     v16l4@u100|:8|:'e--g-b--':||:'f-a-<c-':||:4'g-b-
<d-':|:|
    119: (t1)     @49v10o3l2|:5'da''eb''b-1<d-':|
    120: (t1)     @y$01,$09,$0e¥25|:'da''eb''b-1<d-':|
    121:
    122: (t15)    l16o5r2.re--g-<d-q7c-8>b--8a-8f-8b-<c->b-2..
    123: (t15)    r4rq8>e--g-<d-l8c->b--a-f-b-16<c-16>b-2..
    124: (t15)    >@u80|:'a-4.<c-e-g-'|'b-<d-fa-'l20:|'b-2<d-fa-'|
:'a-16<c-e-g-':|
    125: (t15)    <<rl16v12@u110'fa-'@u127'g-b-'@u100'f4a-'@u110'd
-f'@u120'e-g-'
    126: (t15)    @u95'd-4f'>@u100'b-<d-'@u110'<c-e-'@u100'b-4<d-'
    127: (t15)    @u110'g-b-'@u100'a-<c-'@u90'g-b-'60'g-b-''b-<d-'
'a-<c-''g-b-''fa-'
    128: (t15)    o2l2@u100g-..d-8g-..g-8d-..d-8g-..d-8
    129: (t15)    c-..c-8>b-..b-8a-2.l16@u100a-@u110b-<@u120c-@u12
7cl2d-..@u120d-8
    130: (t15)    g-..d-8g-..g-8d-..d-8g-..d-8c-...c-16>b-..b-8a-4
.a-8<d-g-.c4
    131: (t15)    |:3c-4.c-8c-:|>b-4.b-8<e->l16q6@u127a-4&a-b-<c-c
q8d-2
    132: (t15)    q6>b-4&b-<c-cd-q8e-2l8c-4.c-c2q5d-d-d-d-d-r4.
    133: (t15)    @u127q8g-2&g-d-g-a-l16f4.fe-e--4f4e-4.fe-d-2c-4&
c->b-<c-cd-8.e-8.f8
    134: (t15)    g-2&l8g-d-g-a-l16f4.fe-e--4f4e-4.fe-e--2d-&d-4e-
d->b-<c2
    135: (t15)    l4@u100c-.c-8c-2>b-.b-8<e-2>a-.a-8<d-2>b-.b-8<e-
e-
    136: (t15)    c-.c-8c-2>b-.b-8<e-2>a-.a-8<d-2
    137: (t15)    |:e--.e--8f-2g-.|@u127g-8g-l16g-ff-e-@u100:|@u12
7g-8g-2
    138: (t15)    o2l2@u100g-..d-8g-..g-8d-..d-8g-..d-8
    139: (t15)    c-..c-8>b-..b-8a-2.l16@u100a-@u110b-<@u120c-@u12
7cl2d-..@u120d-8
    140: (t15)    g-..d-8g-..g-8d-..d-8g-..d-8c-...c-16>b-..b-8a-4
.a-8<d-g-.c4
    141: (t15)    @u110|:3c-4.c-8c-:|>b-4.b-8<e->l16q6@u127a-4&a-b
-<c-cq8d-2
    142: (t15)    q6>b-4&b-<c-cd-q8e-2l8c-4.c-c2q5d-d-d-d-d-r4.
    143: (t15)    @u127q8g-2&g-d-g-a-l16f4.fe-e--4f4e-4.fe-d-2c-4&
c->b-<c-cd-8.e-8.f8
    144: (t15)    g-2&l8g-d-g-a-l16f4.fe-e--4f4e-4.fe-e--2d-&d-4e-
d->b-<c2
    145: (t15)    l4@u110c-.c-8c-2>b-.b-8<e-2>a-.a-8<d-2>b-.b-8<e-
e-
    146: (t15)    c-.c-8c-2>b-.b-8<e-2>a-.a-8<d-2g-.g-8g-2c-.c-8c-
2>b-.b-8<e-2
    147: (t15)    >@u115a-.a-8<d-2@u120c-.c-8c-c-8>b-8<@u125a-.a-8
a-.a-8@u127e--.e--8d-l16g-ff-e-
    148: (t15)    |:8r1:||:8l4@u120e--.e--8f-2g-.@u127g-8g-l16g-ff
-e-:|
    149:
    150: /-----------------------------------------------
    151:
    152: (t2)     @i$41,$10,$42
    153: (t2)       i0  @49  o2  v8  l1  q8  p3  @e127,0
    154: (t2)     r1r2g-<v10'e--2g-b--<e--'v11'f-2a-<c-f-''g-b-<d-
g-'
    155: (t2)     v12|:'a-4.<c-e-g-'|'b-<d-fa-'l20:|'b-2<d-fa-'¥20
'a-<c-e-g-'408¥0
    156: (t2)     |:7r1:|l4r2r8@75o5v7@y$01,$63,$58@m30@h0
    157: (t2)     d-8&@b0,-683d-8&@b-683,0d-&@b0d-4.&¥45d-2¥0
    158: (t2)     |:6r1:|@30o4v9@e127,30@m@y$01,$20,$38@y$01,$66,$
60@y$01,$63,$45
    159: (t2)     r2r8.d-16e-*16g-*16@y$01,$63,$40 b-*16@m40@h0<v1
0e-2@m
    160: (t2)     @86o5v5@y$01,$63,$45@y$01,$20,$40@e127,20
    161: (t2)     l16b-<c->b-g-e-c-r8r2rd-fa-b-g-r8r2b-<c->b-fd-8r
8a-4b-8a-g&g2
    162: (t2)     >a-2<d-2'd-1>b-'b-g-e-c-c-e-g-e-b-g-e-cce-g-e-b-
2a-4r4
    163: (t2)     @49o3l2@e127,50v9@y$01,$63,$40@y$01,$66,$40
    164: (t2)     |:'g-1b-<d-''fa-<c-''e--fa-<c-''e-g-b-'|'d-e-g-b
-'
    165: (t2)     'c-e-g-b-''d-fa-':|'g-b-<e--''g-b-<d-''g-b-<c'
    166: (t2)     @53v5l4o4@e127,10|:'c-e-g-b-':||:'c-e-fa-':|>|:'
b-<d-fa-':|
    167: (t2)     |:'b-<d-f-g':||:'a-<c-e-g-':|<|:'c-d-e-g-':||:'>
b-<d-fa-':|
    168: (t2)     'e-g<c-d-''e-gb-<d-'@49>@e100,40|:4'c-e-g-b-':|>
|:'b-<d-fa-':|
    169: (t2)     |:'b-<d-e-g-':||:'a-<c-e-g-':|<|:'c-d-e-g-':|
    170: (t2)     |:'e--g-b--':||:'f-a-<c-':||:4'g-a-b-<d-':|
    171: (t2)     v7|:'e--g-b--':||:'f-a-<c-':||:4'g-a-b-<d-':|
    172: (t2)     @30o3@e127,0l8v8@y$01,$63,$50r¥50'g-2..<d-'¥0|:6
r1:|
    173: (t2)     @101o5v6@e127,55@p74@y$01,$63,$40@y$01,$64,$70q2
r4g-f4e-4d-q8
    174: (t2)     @30o5@e127,0v10@p74@y$01,$63,$40@m
    175: (t2)     |:3r1:|r4.(e,f)6&e*18@y$01,$63,$45g-4a-4@m20¥36g
-1¥0|:3r1:|
```

```
   176: (t2)    @97@y$01,$63,$40@y$01,$64,$40l16p3o5@e127,5v8@m|
:4b-g-e-c-:||:4a-fe-d-:|
   177: (t2)    |:b-g-e-c-:||:a-fe-d-:||:a-fd->b-<:|r2@57o4v15@p
84@e127,20
   178: (t2)    r4g-4¥20e-2¥0r4v15a-4¥25f2¥0
   179: (t2)    @86o5v5p3@y$01,$63,$45@y$01,$66,$60
   180: (t2)    l16b-g-e-c-c-e-g-e-b-g-e-cce-g-e-b-2
   181: (t2)    @30o3@p60v11@y$01,$63,$40@y$01,$66,$40 d-4&(d->f
)48
   182: (t2)    @49o3l2p3@e127,50v9@y$01,$63,$40@y$01,$64,$40
   183: (t2)    |:'g-1b-<d-''fa-<c-''e--fa-<c-''e-g-b-'|'d-e-g-b
-'
   184: (t2)    'c-e-g-b-''d-fa-':|'g-b-<e--''g-b-<d-''g-b-<c'
   185: (t2)    @53v5l4o4@e127,10|:'c-e-g-b-':||:'c-e-fa-':|>|:'
b-<d-fa-':|
   186: (t2)    |:'b-<d-f-g':||:'a-<c-e-g-':|<|:'c-d-e-g-':||:'>
b-<d-fa-':|
   187: (t2)    'e-g<c-d-''e-gb-<d-'v6|:4'c-e-g-b-':|>|:'b-<d-fa
-':|
   188: (t2)    |:'b-<d-e-g-':||:'a-<c-e-g-':|<|:'c-d-e-g-':|
   189: (t2)    |:4'g-<d-':||:4'c-b-':|>|:'b-<a-':||:'b-<g-':|
   190: (t2)    |:'a-<g-':|<|:'c-g-':||:4'c-b-':|>|:4'a-<g-':|<|
:'e--b--':|'c-a-'r4
   191: (t2)    @49@e127,50v6o3l2|:4'e--g-b--''f-a-<c-''g-1a-b-<
d-':|
   192: (t2)    v8<|:8'e--g-b--<g-''f-a-<c-<c-''g-1b-<d-b-':|
   193: (t2)    @e127,80|:5'e--g-b--<g-''f-a-<c-<c-''g-1b-<d-b-'
:|
   194: (t2)    @y$01,$09,$0e¥25|:'e--g-b--<g-''f-a-<c-<c-''g-1b
-<d-b-':|
   195:
   196: /-----------------------------------------------
   197:
   198: (t3)    @i$41,$10,$42
   199: (t3)      i0  @58  o3  v5  l16  q8  @p104  @e127,0
   200: (t3)    r1r1r4g-b--<c-d-@y$01,$63,$30l4v8e--v9f-v10g-v11
a-v12b-2.r
   201: (t3)    @y$01,$63,$40@57l8r2v13@e127,10rb-16r16<d-16v14¥
30f.&f2¥0
   202: (t3)    l16v0rv12fv14g-v12fv14g-v12fd-¥20d-&d-1&d-1¥0r4.
   203: (t3)    @74o5v12l8p3@m30@h24@e127,0b-a-g-fd-c-4.g-16g-4&
g-16rd-c-4
   204: (t3)    b-a-fg-16e-.f16d-16&d-2r4.d-e->g-g-<e-16e-16&e-4
r4fd-d-
   205: (t3)    f16f16&f4r4g-e-e-g-4.g-f&f2r2..d-b-a-g-ff4.g-16g
-16&g-4
   206: (t3)    rd-c-c-16e-.b-a-fe-fe-.d-16&d-4r4.d-
   207: (t3)    e->g-g-<e-16e-4.r16e-fd-d-f16f16&f4rfg-e-16e-.e-
g-4fg-16g-2.&g-16r4
   208: (t3)    v15e-4e-l16d-e-&e-4e-8g-8f4d-8e-d-&d-4r8>a-b-<c-
4c-8e-d-8.
   209: (t3)    e-d-8.c-8d-4d-8c->b-4r<c-8>b-a-4&a-l8a-<g-fe-fe-
16d-4&d-16
   210: (t3)    d-d-16b-.a-g-f16e-4r16e-fg-e-fg-b-g-16a-16&a-4r4
   211: (t3)    r4@57o4@ml8@p104v10@e127,10@y$01,$63,$45
   212: (t3)    |:rd-.>b-.g-.d-.¯2>b-12<¯c-¯b-<|¯e-.f4¯g-¯a-_12:
|
   213: (t3)    v11e-fg-a-|:11r1:|@y$01,$63,$40@y$01,$64,$40
   214: (t3)    @69o5v12l8p3@m30@e127,0r4.b-a-g-fd-c-4.g-16g-4&g
-16rd-c-c-16b-.
   215: (t3)    a-16f.g-16e-.f16d-16&d-2r4.d-e->g-g-<e-16e-16&e-
4re-fd-d-
   216: (t3)    f16f16&f4r4g-e-e-g-4.g-f&f2r2..d-b-a-g-ff4.g-16g
-16&g-4
   217: (t3)    d-d-16c-.c-e-b-16a-.fe-fe-.d-16&d-4r4.d-
   218: (t3)    e->g-g-<e-16e-4.r.fd-d-f16f16&f4rfg-e-16e-.e-g-4
fg-16g-2.&g-16r4
   219: (t3)    v13e-4e-l16d-e-&e-4e-8g-8f8f8d-8e-d-&d-4r8>a-b-<
c-4c-8e-8d-8
   220: (t3)    e-d-8.c-8d-4d-8c->b-4r<c-8>b-a-4&a-l8a-<g-fe-fe-
16d-4&d-16
   221: (t3)    d-d-16b-.a-g-f16e-4&e-16e-fg-e-fg-b-<c-16>b-a-.
   222: (t3)    @30p3@e127,30o3@p60@ma-4&(a-,c-)48
   223: (t3)    o4v8@e127,0@m30@p78'g-1<d-'l2fde-d-c-d-8.e-8.f8
   224: (t3)    g-1f1<e-fl8rv9(f,g-)6&f*18d->v12b-&b-2&@h0@m50¥4
0b-1¥0
   225: (t3)    v10p3@e127,30>r1r1r2.b-&(b-,a-)12&b-16v11¥20g-1&
g-1¥0r1
   226: (t3)    v10<<¥20g-1&g-2¥0r2v11r1r1r4.@ml16d-g->f2g-2.{rg
-a-}4
   227: (t3)    (a-,b--)6&a-*18(a-,b--)6&a-*18&a-8{b--a-g-}8@m40
a-2¥70g-1¥0
   228: (t3)    r1v9r2...@y$01,$63,$43 a-8&(a-,b-)36&¥35a-2&a-4¥
0
   229: (t3)    r1r4.<v9@y$01,$63,$4cg-8&¥38g-1¥0r2
   230: (t3)    l2@31o2@y$01,$63,$20@y$01,$64,$60@y$01,$66,$50v5
q7|:8'g-8<d-'¯11:|
   231: (t3)    q8v10@y$01,$63,$40@y$01,$64,$40@y$01,$66,$40
   232: (t3)    'da''eb''g-1<d-''da''eb''g-2..<d-'|:'d16a':|
   233: (t3)    'da''eb''g-1<d-''da''eb''g-4.<d-'r8v8|:8'b-16d-g
-':|
   234: (t3)    |:'da''eb''g-1<d-':|'da''eb''g-4.<d-'r8
   235: (t3)    |:4'g-16b-''g-16<c-':|'da''eb'|:16'g-16b-':|
   236: (t3)    v11d4&(d>a)48_3(g-,b)12&g-4..v9¥20b-1¥0
   237: (t3)    v12o6@30@y$01,$63,$50r1r1r4@b0,2049d-8&@m80d-4&¥
60d-4&@m@b2049,-4098d-2¥0
   238: (t3)    @b0r2..r1i8@32o4v11@y$01,$63,$35@e70,0g-2&g-8.&v
8(g-,c-)12r1r2.
   239: (t3)    i0@75o5v8@y$01,$63,$58@m50¥20@e127,0r2d2.&(d,d-)
48&|:15d16&_5:|d16
   240:
   241: /-----------------------------------------------
   242:
   243: (t4)    @i$41,$10,$42
   244: (t4)      i0  @49  o4  v5  l16  q8  @p44  @e50,0
   245: (t4)    r1r1r4g-v7b--<v8c-v9d-e--4&e--e--g-<d-l8c->b--a-
f-b-16<c-16>b-2..
   246: (t4)    o5l8p3v9a-<c-e->b-4<fd-4>a-<c-e->b-4<fd-d-16¥18d
-16&d-1&d-1¥0
   247: (t4)    |:8r1:|o2@e127,80v8g-*312a-b-<c-d-1&l4d-e-g-f
   248: (t4)    v8g-1&g-a-g-d-d-1&d-c->b-a-<<l8e-4.d-16e-16&e-4
   249: (t4)    rg-f4d-e-16d-4..>b-<c-4.e-d-e-16d-.c-d-4.c-16>b-
16&b-2
   250: (t4)    <v9@e127,50e-2.l4v10fa-g-fg-e-fg-a-b-2a-r
   251: (t4)    @30o4v8@e127,0@m30@p78'g-1<d-'l2fde-d-c-d-8.e-8.
f8
   252: (t4)    g-1f1<e-fl8rv9(f,g-)6&f*18d->v12b-&b-2&@h0@m35¥3
8b-1¥0
   253: (t4)    v8|:6r1:|<@y$01,$63,$45¥35g-1¥0r1v10@y$01,$63,$5
0
   254: (t4)    @p74r*12@m20¥35g-1&g-*180¥0p3|:7r1:|l4r2r8
   255: (t4)    @75o5v7@y$01,$63,$54@m30@h0d-8&@b0,-683d-8&@b-68
3,0d-&@b0d-4.&¥45d-2¥0
   256: (t4)    |:7r1:|@49@y$01,$63,$40@y$01,$66,$45v10@e127,30q
5l16@m
   257: (t4)    r2b-<c->b-g-e-c-r8r2rd-fa-b-g-r8r2b-<c->b-fd-8r8
   258: (t4)    q8a-4b-8a-g&g2>a-2<d-2'd-1>b-'
   259: (t4)    v6b-g-e-c-c-e-g-e-b-g-e-cce-g-e-v10b-2a-4r4
   260: (t4)    @57o4l8@p104v10@e127,10@y$01,$63,$45@y$01,$66,$4
0
   261: (t4)    |:rd-.>b-.g-.d-.¯2>b-12<¯c-¯b-<|¯e-.f4¯g-¯a-_12:
|
   262: (t4)    v11e-fg-a-@49o4l4p3v11@e127,40@y$01,$63,$40@y$01
,$66,$45
   263: (t4)    re-fe-d-1rc-d-e-f2'g<c-''gb-'q3l8v9@e127,80
   264: (t4)    <b-b-g-g-e-e-c-c-a-a-ffd-d->b-b-q8<v8g-2v9a-2
   265: (t4)    v8'g-1b-'q3v9b-b-g-g-e-e-c-c-a-a-ffd-d->b-b-
   266: (t4)    <l2q8g-a-'c-.b-'')b-4<<c-'')a-1<<d-1''e--<e--''d
-<d-'
   267: (t4)    @100o5v7p3@e80,10@y$01,$63,$40@y$01,$66,$30@y$01
,$64,$30
   268: (t4)    |:17r4l16re--g-<d-l8c->b--a-f-b-16<c-16>b-2..:|
   269: (t4)    ¥25|:r4l16re--g-<d-l8c->b--a-f-b-16<c-16>b-2..:|
   270:
   271: /-----------------------------------------------
   272:
   273: (t5)    @i$41,$10,$42
   274: (t5)      i0  @36  o2  v14  l8  q8  p3  @e127,0
   275: (t5)    r1r4r2..r4g-16d-16e--2f-2g-.d-.>g-2&g-
   276: (t5)    a-4.b-*120a-4.b-2<d-16d-1&d-1&d-16
   277: (t5)    |:16r1:|v13c-4.c-c-4.b-16<c-16>c-4.
   278: (t5)    c-c-4b-16<c-16>g-c-4.c-c-4.c->b-4.b-<e-2
   279: (t5)    >l16q7a-4&a-b-<c-cq8d-2>q7b-4&b-<c-cd-q8
   280: (t5)    e-4.l8v12d-c-4.c-c4.c16c16q5d-d-d-d-q8
   281: (t5)    @30@e127,20o3@p60v11d-4&(d->f)48
   282: (t5)    @36o2v13@e127,0p3g-4.g-g-d-g-a-f4.
   283: (t5)    l16fe-e--4f4e-4.fe-d-4.d-8c-4&c->b-<c-cd-8.e-8.f
8
   284: (t5)    g-4.l8g-g-d-g-a-f4.l16fe-e--4f4e-4.fe-e--2d-4&d-
e-d->b-<c2
   285: (t5)    |:11r1:|o1l4v10|:g-.g-8g-.<d-8|:g-.g-8:||:d-.d-8
:|
   286: (t5)    >g-.g-8g-<d-8>g-8<|:c-.c-8:|>|:b-.b-8:|a-.a-8|a-
2<d-2..d-8:|
   287: (t5)    <d-2>g-.g-8g-<c8&(c>c)24v11l8c-4.c-c-4.b-16<c-16
>c-4.
   288: (t5)    c-c-4b-16<c-16>g-c-4.c-c-4.c->b-4.b-<e-2
   289: (t5)    >l16q7a-4&a-b-<c-cq8d-2>q7b-4&b-<c-cd-q8
   290: (t5)    e-4.l8v12d-c-4.c-c4.c16c16q5d-d-d-d-q8d-4r4
   291: (t5)    o2v11g-4.g-g-d-g-a-f4.l16fe-e--4f4e-4.fe-d-4.
   292: (t5)    d-8c-4&c->b-<c-cd-8.e-8.f8
   293: (t5)    g-4.l8g-g-d-g-a-f4.l16fe-e--4f4e-4.fe-e--2d-4&d-
e-d->b-<c2
   294: (t5)    l4v11c-.c-8c-2>b-.b-8<e-2>a-.a-8<d-2>b-.b-8<e-e-
c-.
   295: (t5)    c-8c-2>b-.b-8<e-2>a-l16a-b-<c-cd-2
   296: (t5)    >l8g-4.g-<a-g-d->g-<c-4.c-c-2>b-4.b-<e-2
   297: (t5)    l16>a-4&a-b-<c-cd-4l8d-cc-4.c-c-4c->b-
   298: (t5)    a-4.a-a-2<e--4.e--d-4d-4|:7r1:|>
   299: (t5)    v8|:8g-¯5:|v12<e--4.e--f-4&f-16f-16c-16>g-16
   300: (t5)    g-4.<d-g-d->g-f-<e--4.e--f-4&f-16f-16c-16>f-16
   301: (t5)    l4g-&{g-<e-d-}{>b-a-g-}{ff-e-}
   302: (t5)    <e--.e--8f-&f-16l16f-c->f-g-4..<d-g-g-8d-g-8
   303: (t5)    ee-e--4.e--8f-4&f-f-c->f-g-g-g-a-a-a-b-b-b-<c-c-
c-d-d-f-f-
   304: (t5)    e--e--d-->e--<e--8e--8f-f-c->f-<f-8>f-8
   305: (t5)    g-g-8<d-g-8g-g-8d->g-8g-ff-e-e--e--8b--
   306: (t5)    <e--8>e--8f-f-8<c-f-8>f-8g-g-g-a-a-a-b-b-b-
   307: (t5)    <c-c-c-d-d-f-f-e--e--8>e--<e--8e--8f-f-8>f-
   308: (t5)    <f-8f-8|:5v14fv12g-g-:|v13g-e--4.e--8f-4&f-f-c->
f-
   309: (t5)    l8v14g-4.<d-g-f->g-f-e--1v12¥20g-1¥0
   310: (t5)    |:5r1:|@33@y$01,$64,$60<v12e1v14@y$01,$64,$55 d1
v13¥30e1
   311:
   312: /-----------------------------------------------
   313:
   314: (t6)    @i$41,$10,$42
   315: (t6)      i0  @64  o2  v8  l4  q8  p3  @e127,0
   316: (t6)    @y$01,$63,$5ar1r2g-1|:r1:|@31o2v11l4@y$01,$63,$4
0@e127,10@m80@h10
   317: (t6)    'a-.<e-''b-<f'120'a-.<e-''b-2<f'@m30<v14'e-16a-'
@h0¥18'e-a-'348¥0r4
   318: (t6)    |:13r1:|r1r1@31o3v12p3@e127,50@y$01,$66,$60@y$01
,$63,$45
   319: (t6)    r2r8.d-16e-*16g-*16@y$01,$63,$40v10(a-,b-)10&a-*
6@m40@h0<v12e-1&e-1
   320: (t6)    r1r1@y$01,$66,$40@m@57o4v14@p94@e127,20
   321: (t6)    r4g-4¥30e-2¥0v15r4a-4¥30f2¥0
   322: (t6)    @30p3@e127,20@y$01,$63,$40r1r2o3@p60a-4&(a-,c-)4
8
   323: (t6)    @49o6l4v8@e80,0q1@p84d-g-a-g-a-g-fa-e-g-d-g-a-g-
fa-
   324: (t6)    d-g-b-g-a-v7g-fa-a-g-fv6g-v5a-g-v3fg-|:4r1:|@49o
5q3l8v7@e127,80p3@m
   325: (t6)    b-b-g-g-e-e-c-c-a-a-ffd-d->b-b-q8v6<g-2@y$01,$66
,$43 a-2r4@y$01,$66,$40>
   326: (t6)    l16v10@e127,20re--g-<d-l8c->b--a-f-b-16<c-16>b-2
```


▶ゲームミュージックライブに行ってきました。時代はやはり兄貴です。当然，今一番好きなものは「プロ，プロ，プロテイン」です。　　信太　徹(23)神奈川県

```
..¯4116
   327: (t6)     r4re--g-<d-18c->b--a-f-b-16<c-16>b-2&b-r4
   328: (t6)     o314v9@e127,40|:3g-a-b-<c-d-e-fg->:||g-a-b-<c-d-
1
   329: (t6)     v8|:4g-a-b-<c-d-e-fg->:|
   330: (t6)     @86o5v518@p60@y$01,$63,$45@y$01,$66,$55@e127,60
   331: (t6)     18e-4.d-16e-16&e-4rg-f4d-e-16d-4..>b-<c-4.e-d-e-
16d-.c-
   332: (t6)     d-4.c-16>b-16&b-214<e-2.fa-g-fg-e-fg-a-b-2a-r
   333: (t6)     @49o614v8@e80,0q1@p84@y$01,$63,$40@y$01,$66,$40
   334: (t6)     d-g-a-g-a-g-fa-e-g-d-g-a-g-fa-d-g-b-g-a-v7g-fa-a
-g-fv6g-v5a-g-v3fg-
   335: (t6)     |:7r1:|@57o418v13@e127,30q8@y$01,$63,$41
   336: (t6)     r4b-4¯2<c-.>¯b-.<¯c-¯d-2.¯e-4v15d-1
   337: (t6)     r114v13r>e-fg-¯a-¯b-<¯c-¯d-v15e--2v14d-2
   338: (t6)     @1o2v16p3@e100,10@y$01,63,$40
   339: (t6)     |:414@u100e--.e--8f-2g-.@u120g-8g-116g-ff-e-:|
   340: (t6)     @62o4v14@p44@e127,0q7@y$01,$64,$60@y$01,$66,$55@
y$01,$63,$30
   341: (t6)     18|:8r1:||:4g-.b--.<f-e-16d-16c-4.>b-16<c-16>b-2
..:|
   342: (t6)     q5|:5g-<d-16>b--.<f-e-16d-16c->r1r4:|
   343: (t6)     ¥25|:g-<d-16>b--.<f-e-16d-16c->r1r4:|
   344:
   345: /----------------------------------------------
   346:
   347: (t7)     @i$41,$10,$42
   348: (t7)       i0  @44  o5  v15  18  q8  p3  @e127,0
   349: (t7)     |:31r1:|r1r4g-a-16b-4..d-<c->b-a-g-a-4&a-16r16b-
16b-.<c->b-a-16a-16
   350: (t7)     g-4.g-16g-.e-fg-b-4.g-16a-4r16g-a-16
   351: (t7)     b-4..d-<c->b-a-b-16a-4&a-16r8fb-4a-4a-g-fg-&g-1r
4v12@74e-f
   352: (t7)     g-4rg-a-fe-e-e-.d-16&d-4r4c->b-<c-4rg-f.e-.d-b-2
<c-4>b-4
   353: (t7)     14b-g-8e-16g-16&g-ra-f8a-16g-16&g-r8g-8e-g-18<c-
.>b-.g-16g-16&¥30g-1&g-1¥0
   354: (t7)     |:10r1:|@49o2@e127,80v10g-*312a-b-<c-d-1&14d-e-g
-f
   355: (t7)     g-1&g-a-g-d-d-1&d-c->b-a-|:7r1:|
   356: (t7)     @44o5v15@e127,018r2.g-a-16b-4..d-<c->b-a-g-a-4&a
-16r16b-16b-.<c->b-a-16a-16
   357: (t7)     g-4.g-16g-.e-fg-b-4.g-16a-4r16g-a-16
   358: (t7)     b-4..d-<c->b-a-b-16a-4&a-16r8fb-4a-4a-g-fg-&g-1r
4v14@69e-f
   359: (t7)     g-4rg-a-fe-e-e-.d-16&d-4r4c->b-<c-4rg-f.e-.d-b-2
<c-4>b-4
   360: (t7)     14b-g-8e-16g-16&g-ra-f8a-16g-16&g-r8g-8e-g-18<c-
.>b-.g-16g-16&g-2
   361: (t7)     @44@e127,10r.<c-.>b-b-4g-e-16g-16&g-4r4a-4<d->a-
16g-16&g-4rg-
   362: (t7)     e-4g-4<c-.>b-.g-16g-2..a-¥30g-1&g-1¥0
   363: (t7)     |:6r1:|r1r2..@44o5v16@y$01,$09,$0e@y$01,$66,$48
   364: (t7)     g-*25g-*11g-*18&g-1&g-1&g-2..&¥26@m50@h0g-2¥0@mr
*6
   365: (t7)     v14d-4&(d-<d-)48&v16@m50¥20>d-2&d-2.¥0r16@m
   366: (t7)     v15<b-*24b-*16b-*20v14d-8.d-16.d-16.d-8.r2r8
   367: (t7)     116v15d-8c->|:b-<c-8>:|g-g-8g-g-8g-v16
   368: (t7)     116q7|:3g-f-g-|g-8r8.:|@y$01,$09,$40g-2.rd-8c->v
15b--<v16q8g-1&g-4r8.
   369: (t7)     q7g-8rg-8rg-8g-4116re--g-<d-18c->b--a-f-b-16<c-1
6>b-2&b-8.
   370: (t7)     <d-*30d-*10d-1&d-1@31o2v12@e127,10
   371: (t7)     a4&(a,d)48_3(c-,e)12&c-4..v9(e-,g-)12&¥20e-2...¥
0
   372:
   373: (t8)     o418@u115
   374: (t8)     |:32r1:||:r4b-<c-16d-4..>b-<e-d-c->b-<c-4.d16d.d
dd16d16>b-4.b-16b-.
   375: (t8)     g-a-b-<d-4.>b-16<c-16&c-4>b-<c-16
   376: (t8)     d-4..>b-<e-d-c-d-16c-4..>b-<d4f4fe-de-&e-1|r2|:3
4r1:|r2:|
   377:
   378: /----------------------------------------------
   379:
   380: (t10)   @i$41,$10,$42
   381: (t10)     i0  @1  o2  v16  14  q8  p3  @e127,0
   382: (t10)   @y$1a,36,$60@y$1a,51,$35@y$1a,42,$40
   383: (t10)   @y$1c,51,$40@y$18,51,$42@y$1a,46,$40
   384: (t10)   @y$1a,38,$70@y$1a,43,$50@y$18,57,$41
   385: (t10)   @y$1a,57,$40@y$1a,47,$50@y$1a,43,$55
   386: (t10)   @y$1a,50,$55@y$1c,47,$40@y$1c,43,$01
   387: (t10)   @y$1c,50,$7e@y$1a,44,$45@y$1a,37,$50
   388: (t10)   @y$1c,37,$40@y$1c,42,$50@y$1c,44,$50
   389: (t10)   @y$1c,46,$50@y$18,52,$45@y$1a,52,$4c
   390: (t10)   @y$1c,52,$40@y$18,48,$3c@y$1c,48,$50
   391: (t10)   @y$1a,48,$50@y$18,45,$3c@y$1c,45,$40
   392: (t10)   @y$1a,45,$50@y$18,41,$3c@y$1c,41,$30
   393: (t10)   @y$1a,41,$50@y$18,42,$43@y$18,42,$43
   394: (t10)   r1r2...r2c16crcr8.c16cr.d8d8d16d16
   395: (t10)   cr8c8r8c16c16rcr8c8r.c16c16
   396: (t10)   |:18r1:||:7cr8c8cr:|c8g8g8g8gr
   397: (t10)   |:|:3cr8c8cr:|c18rc|c.c16rc14:|14cr
   398: (t10)   |:10r1:|r2r8v15c16c16v14r|:|:7cr8c8cr:||cr2.:|
   399: (t10)   c.c8cr|:7c.c8cr:|18gggg116dd<dd>bbgg
   400: (t10)   14|:|:3cr8c8cr:|c18rc|c.c16rc14:|14cr
   401: (t10)   v15|:21c.c8cr:|c8|:5g8:|116'g<d'<d>gg
   402: (t10)   14|:7c.c8cr:|@y$1a,36,$70 ccccc.c8crc.c8ccc.c8cr
cccc
   403: (t10)   |:8c:|@y$1a,36,$70|:8c8:|116c@y$1a,38,$70|:<c>af
'da'<c>af|a:|
   404: (t10)   r1r1r1r4b8bb8gg8gr1r2r8<d8dd>bbr1r2...r8c8c8
   405:
   406: (t11)   r1r1r214o2f+<d+>f+<d+d+8.d+8.d+>g8g8g16g16
   407: (t11)   <a>d8<a8r>g<a>d8<a8.116ddd>bg<aa
   408: (t11)   >|:17r1:|r2.f+f+a+814|:7rf+rd:|r8d8d8d8d<d16>b16
g8
   409: (t11)   @y$1a,42,$5018|:<a>f+'df+'f+|:3f+f+'df+'f+:|
   410: (t11)   <a>f+'df+'f+|:f+f+'df+'f+:||<a.a16>d<a>:|
   411: (t11)   f+<d>b16b16g|:10r1:|r2.d64d8...
   412: (t11)   |:|:7|:6f+:|'df+'f+:||f+4f+f+rf+rf+:|
   413: (t11)   |:6f+:|d16f+16a+|:7g+f+'df+'f+:|f+f+dd16d16
   414: (t11)   |:6g+f+'df+'f+:|ddddr2
   415: (t11)   v15|:<a>f+'df+'f+|:3f+f+'df+'f+:|
   416: (t11)   <a>f+'df+'f+|:f+f+'df+'f+:||<a.a16>d<a>:|
   417: (t11)   f+f+'d16f+'f+16a+<a>f+'df+'f+|:6f+f+'df+'f+:|f+f
+dd16d16
   418: (t11)   |:7f+f+'df+'f+:|f+16'd16f+'f+df+<a>f+'df+'f+
   419: (t11)   |:4f+f+'df+'f+:|f+f+dd16d16<a>f+'df+'f+|:4f+f+'d
f+'f+:|
   420: (t11)   f+<d>b16b16g|:28'c+f+'f+:|f+d|:3'f+d'd:|
   421: (t11)   <a>f+'df+'f+|:3f+f+'df+'f+:|
   422: (t11)   <a>f+'df+'f+|:f+f+'df+'f+:|r16<d16>r16b16'gd'b16
'b16<a'
   423: (t11)   v16<a>f+'df+'f+|:f+f+'df+'f+:|f+f+'d16g'g16b16b1
6<a>f+'df+'f+f+f+'df+'f+
   424: (t11)   116r<'de'r>b'd8<e'b'b<e'r'dg<e''dg<e'<e>'d8g<e''
dg<e''dg<e'18
   425: (t11)   <a>f+'df+'f+f+f+'df+'f+f+f+'df+'f+116r<d>bbd8gg
   426: (t11)   18<a>f+'df+'f+f+f+'df+'f+116r|:3'b<d':|d8|:'bd':
|
   427: (t11)   |:3'g8d':||:'gd':|18<a>f+'df+'f+f+f+'df+''f+16<d
'b16
   428: (t11)   f+f+'d16f+<d'b16'f+g'f+f+116'df+'<e>f+<e>
   429: (t11)   18<a>f+'df+'f+f+f+'df+'f+116<'e8>f+'>f+8<'e8>f+'
   430: (t11)   'e>f+'e>f+<e'e8>f+''e8>f+''e>f+'e@y$1a,52,$44
   431: (t11)   a2.a4r4.dd>b8bgg<eeee8r2..r16a8aa8.r1rar2er4.d>g
r4<
   432: (t11)   |:@y$1a,57,$21@y$1a,52,$10'ae'@y$1a,52,$17@y$1a,
57,$32
   433: (t11)   'ae'@y$1a,52,$1e@y$1a,57,$3c'ae'@y$1a,57,$41@y$1
a,52,$25'ae'
   434: (t11)   @y$1a,57,$46@y$1a,52,$2c'ae'@y$1a,57,$50@y$1a,52
,$3b'a8.e':|
   435: (t11)   r2r8>@y$1a,52,$4c@y$1a,43,$65@y$1a,57,$40
   436: (t11)   g8@y$1a,43,$40r2<dd>b8bgg<eeea8r2r8a8@y$1a,52,$4
3
   437: (t11)   r2dd>b8bgg<eeee8¥25r2..ra8a'a8>b'>bb8gg8g8
   438:
   439: /----------------------------------------------
   440:
   441: (t17)     @2  o2  v7  14  q8  p3  y48,00  y15,00
   442: (t17)   r2|:22r1:|r2.r2...d-16e-*16g-*16(a-,b-)10&b-*6<e
-1&e-1>
   443: (t17)   r1r1rv8o2g-e-2ra-f2r1r2o3a-.&(a-,c-)24(e-,g-)24|
:38r1:|r2..
   444: (t17)   o214rg-e-2ra-f2r1r2o3a-.&(a-,c-)24(e-,g-)24|:21r
1:|r2..
   445: (t17)   @3o6|:13r4116re--g-<d-18c->b--a-f-b-16<c-16>b-8r
2.:|
   446: (t17)   v7|:4r4116re--g-<d-18c->b--a-f-b-16<c-16>b-8r2.:
|
   447: (t17)   ¥28|:r4116re--g-<d-18c->b--a-f-b-16<c-16>b-8r2.:
|
   448:
   449: /----------------------------------------------
   450:
   451: (t18)     @2  o2  v7  14  q8  p3  y49,40
   452: (t18)   r2|:22r1:|r2.r2...d-16e-*16g-*16(a-,b-)10&b-*6<e
-1&e-1>
   453: (t18)   |:5r1:|r2>a-.&(a-,c-)24(e-,g-)24|:38r1:|r2..
   454: (t18)   @2o2v8rg-e-2ra-f2r1r2o3a-.&(a-,c-)24(e-,g-)24r2.

   455: (t18)   |:28r1:|18|:8g-¯2:|12v12deg-1deg-..d16d16
   456: (t18)   deg-1deg-4.r8|:8b-16:||:deg-1:|deg-4.r8v8|:8d-16
:|v10de|:16g-16:|
   457: (t18)   v13d4&(d>a)48v10(g-,b)12&g-4..v9(b-<d-)12&>¥20b-
2...¥0
   458:
   459: /----------------------------------------------
   460:
   461: (t25)     o2  v10  14  p3
   462: (t25)   r1r2r2...c+16c+rc+r8.c+16c+rr8d8d8d16d16
   463: (t25)   cd8cc16c16rcd8c2c16c16
   464: (t25)   |:18r1:||:7c.c8cd:|18cddd14dr
   465: (t25)   |:|:3cd8c8cd:|cd8c8|c8.c16d8c8:|cr
   466: (t25)   |:10r1:|r2r8c+16c+16d+
   467: (t25)   |:|:7c+.c+8c+d+:||cr2.:|
   468: (t25)   c.c8cd|:4cd8c8c|d:|d8d16d16
   469: (t25)   |:3cd8c8cd:|18ddddr2
   470: (t25)   14|:|:3cd8c8cd:|cd8c8|c8.c16d8c8:|cd
   471: (t25)   |:|:3c+d+8c+8c+d+:||c+d+8c+8c+d+8d+16d+16:|
   472: (t25)   18c+4d+c+c+16d+.d+4|:c+4d+c+c+4d+4:|
   473: (t25)   14c+d+8c+8c+d+8d+16d+16|:c+d+8c+8c+d+:|c+d+8c+8c
+2
   474: (t25)   |:7r1:|c8|:7d8:||:7cd8c8cd:|ccc16d16d8c8d16d16
   475: (t25)   cd8c8cdcd8c8cccd8c8cdcccc|:8c:||:8c+8:|
   476: (t25)   cdcd|:7r1:|r2..c8c8
   477:
   478: /----------------------------------------------
   479:
   480: (p)
   481:
   482: .sc55_print "Dreams Come True"
   483: .comment   LOVE GOSE ON …より  ～未来予想図II～
```

リスト2　未来予想図Ⅱの音色コンフィグファイル

```
.o2d = c:sd.pcm,v50
.o2c = c:bd.pcm,v100
.o2d+= c:sd.pcm,v40
.o2c+= c:bd.pcm,v70
```

▶裕木奈江さんっていいですよね。同好の士，求む！　　吉岡　昌樹(18)埼玉県

リスト3　未来予想図Ⅱのカウンタ表示

```
 1:000065A0 00000000   2:000065A0 00000000   3:00006450 00000000   4:000065A0 00000000
 5:000062A0 00000000   6:000065A0 00000000   7:00005CA4 00000000   8:00003E40 00000000
15:00005B20 00000000  10:00006138 00000000  11:00006408 00000000  17:000065A0 00000000
18:00005CA0 00000000  25:00006138 00000000
```

リスト4　PASSING BREEZE

```
     1: .comment OUT RUN - PASSING BREEZE - PCM8 By TTN 93/07/26
     2: .comment                    Special thanks to 阿蒼 93/05/
05～
     3: .ADPCM_BLOCK_DATA = PASSING.ZPD
     4: /-------------------------------------------------------
     5: /  PASSING.ZMS  v2.14
     6: /
     7: /  for X680x0 PCM8.X & ZMUSIC.X
     8: /
     9: /  Out Run (c)SEGA 1986 「Ｐａｓｓｉｎｇ　Ｂｒｅｅ
ｚｅ」
    10: /
    11: /  Programmed by 舘野 暢/TTN  1993/07/26
    12: /
    13: /-------------------------------------------------------
    14:
    15:
    16: /-------------------------------
    17: / TRACK SETUP
    18: (i)
    19:
    20: (m1,2000)(aFM1,1)
    21: (m2,2000)(aFM2,2)
    22: (m3,2000)(aFM3,3)
    23: (m4,2000)(aFM4,4)
    24: (m5,2000)(aFM5,5)
    25: (m6,2000)(aFM6,6)
    26: (m7,2000)(aFM7,7)
    27: (m8,2000)(aFM8,8)
    28: (m9,1000)(aADPCM,9)
    29: (m10,1000)(aADPCM,10)
    30: (m11,1000)(aADPCM,11)
    31:
    32:
    33: /-------------------------------
    34: / OPM DATA SET
    35:
    36: /     AR  DR  SR  RR  DL  TL  KS  MT DT1 DT2
    37: / BASS
    38: (@01, 31, 31,  0,  9,  0, 28,  0,  0,  0,  0,  0
    39:       31, 12, 12,  9,  8, 43,  0,  8,  0,  0,  0
    40:       31, 14, 10,  9,  3, 21,  0,  0,  0,  0,  0
    41:       31, 11,  8,  9,  1,  4,  0,  1,  0,  0,  0
    42: /     AL  FB  OM
    43:        2,  2, 15)
    44: / CHORD 1
    45: (@02, 28,  1,  7,  9,  0, 28,  0,  1,  0,  0,  0
    46:       30,  1,  8, 10,  0,  1,  0,  1,  0,  0,  0
    47:       29,  1,  8,  9,  0, 19,  0,  1,  3,  0,  0
    48:       31,  1,  8, 10,  0,  1,  0,  1,  3,  0,  0
    49:        4,  7, 15)
    50: / CHORD 2
    51: (@03, 31,  0,  0,  5,  0, 29,  0,  1,  3,  0,  0
    52:       31,  0,  0,  5,  0, 43,  0,  1,  3,  0,  0
    53:       31, 17,  0,  5,  6, 49,  0, 15,  0,  0,  0
    54:       31,  9,  8,  9,  2,  1,  0,  1,  0,  0,  0
    55:        3,  7, 15)
    56: / MAIN
    57: (@04, 20,  9,  0,  9,  5, 27,  2,  8,  0,  0,  0
    58:       23,  0,  0, 10,  0,  1,  0,  8,  0,  0,  0
    59:       18,  5,  0,  9,  1, 17,  0,  4,  1,  0,  0
    60:       22,  0,  0, 10,  0,  2,  0,  4,  1,  0,  0
    61:        4,  7, 15)
    62:
    63:
    64: /-------------------------------
    65: / MML DATA SET
    66: (o127)
    67:
    68: /---------------
    69: / CHORD
    70:
    71: (t1)  @02o4@k01 v13‾1L16 q8
    72: (t1)  p3|:e1^4^16er16p3er4f+4
    73: (t1)  g1^4ggr16gr4|f4:|f4
    74: (t1)  |:7rc+:|rc+e1r8 @v0r
    75: (t1)  @03o5 v13‾1p3e4>e4>e4^16
    76: (t1)  |:@02o4 v13‾1 L4
    77: (t1)  p3|:e1^2g+ee1^2g+ed1^2ef+d8.^2.d16^1:|L8
    78: (t1)  p3|:|:e.e16r2:|e.e16r8 e^1r8<|
    79: (t1)   e16e16e16e16r8e.e16r4>:|
    80: (t1)  <e16r8e16r8e4r4>
    81: (t1)    |:|:e.e16r2:|e.e16r8|e^1r8<
    82: (t1)   e16e16e16e16r8e.e16r4>:|
    83: (t1)  e^1^4er8p3 e.d+.d^2.^8.e16^2.^8.f+16^1L16
    84: (t1)  <p1r16f+r16p2f+r16p1f+r16p2f+r16r4.p3>
    85: (t1)  p3L1f+ee2.^8.^16 f+16^4..e2 f+ee^2^16e4..
    86: (t1)  f+ee4^8^16^16 f+16^4^8.e16^4^4 r4..
    87: (t1)  f+ef+^16 L12rf+f+f+f+f+f+r4rr:|
    88: (t1)  |:@03p3o5 e1^4d2g8.d16^2.>b8.<d16^1
    89: (t1)  e2.d8.e16^2f4..d16^2.g8.d16^1
    90: (t1)  e1^2d4e8.d16^1^1
    91: (t1)  e1^2d4e8.d16^1<d4..d16^2:| @02o4
    92: (t1)  L1p3|:5f+ee f+2e2:| f+e;$cc,6e f+2e2 |:7f+ee f+2e2
:|
    93:
    94:
    95: (t2)  @02o4@k01 v13‾1L16 q8
    96: (t2)  p3|:c+1^4^16c+r16c+r4c+4
    97: (t2)  e1^4eer16er4|d4:|d4>
    98: (t2)  |:7ra:|rab1r8. @v0r
    99: (t2)  @03o5 v13‾1p3c+4>c+4>c+4
   100: (t2)  |:@02o4 v13‾1 L4
   101: (t2)  p3|:c+1^2c+c+c+1^2c+c+c+1^2c+c+>b8.^2.b16^1<:|L8
   102: (t2)  p3|:|:c+.c+16r2:|c+.c+16r8 c+^1r8<|
   103: (t2)   c+16c+16c+16c+16r8c+.c+16r4>:|
   104: (t2)  <c+16r8c+16r8c+4r4>
   105: (t2)    |:|:c+.c+16r2:|c+.c+16r8|c+^1r8<
   106: (t2)   c+16c+16c+16c+16r8c+.c+16r4>:|
   107: (t2)  c+^1^4c+r8c+.c.>b^2.^8.b16^2.^8.<c+16^1L16
   108: (t2)  <r16p1c+r16c+r16p1c+r16p2c+r16r4.r*4>
   109: (t2)  p3L1ec+c+2.^8.^*8c+16^4..c+2^*4 ec+c+^*8^6^6c+6^4.
.^*4
   110: (t2)  ec+c+4^8.^*8c+16^4^8.c+16^4^8.c+2^*4
   111: (t2)  ec+c+^*8 L12rc+c+c+c+c+c+rrrrr:|
   112: (t2)  |:@03p3o5 c1^4c2c8.>b16^2.^8.b16^1<
   113: (t2)  c2.c8.c16^2c4..>b16^2.^8.b16^1<
   114: (t2)  c1^2c4c8.>b16^2^8r4b8^1<
   115: (t2)  c1^2c4c8.>b16^2r4.<b8^*88b*8^2:| @02o4
   116: (t2)  L1p3|:13ec+c+2^*92 c+2c+2^*4:|
   117:
   118:
   119: (t3)  @02o3@k01 v13‾1L16 q8
   120: (t3)  p1|:b1^4^16br16br4b4<
   121: (t3)  c1^4ccr16er4|c4:|c4>
   122: (t3)  |:7rf+:|rf+g+1r4 @v0r
   123: (t3)  @03o4 v13‾1p1b4>b4>b8.
   124: (t3)  |:@02o3 v13‾1 L4
   125: (t3)  p1|:b1^2bba1^2aaa1^2aaa8.^2.g+16^1:|L8
   126: (t3)  p1|:|:a.a16r2:|a.a16r8 b^1r8<|
   127: (t3)   b16b16b16b16r8b.b16r4>:|
   128: (t3)  <b16r8b16r8b4r4>
   129: (t3)    |:|:a.a16r2:|a.a16r8|b^1r8<
   130: (t3)   b16b16b16b16r8b.b16r4>:|
   131: (t3)  b^1^4br8b.a+.a^2.^8.g+16^2.^8.a+16^1L16
   132: (t3)  <r16p1a+r16p2a+r16p1a+r16p2a+r16r4.r*8
   133: (t3)  p1L1c+>ba2.^8.^*4a16^4..a2^*8< c+>ba^*4^6a3^4..^*8
   134: (t3)  <c+>ba4^8.^*4a16^4^8.a16^4^8.a2^*8
   135: (t3)  <c+>ba+^*4 L12ra+a+a+a+a+a+r4rr:|
   136: (t3)  |:@03p1o4 a1^4a2a8.g16^2.^8.g16^1
   137: (t3)  a2.a8.a16^2a4..g16^2.^8.g16^1
   138: (t3)  a1^2a4a8.g16^2^8r8g4^1
   139: (t3)  a1^2a4a8.g16^2r4<g4^*92g*4^2:| @02o4
   140: (t3)  L1p1|:13c+>ba2^*88 a2a2^*8:|
   141:
   142:
   143: (t4)  @02o3@k01 v13‾1L16 q8
   144: (t4)  p2|:g+1^4^16g+r16g+r4g+4
   145: (t4)  a1^4aar16ar4|a4:|a4
   146: (t4)  |:7rf+:|rf+e1r4^16 @v0r
   147: (t4)  @03o4 v13‾1p2g+4>g+4>g+8
   148: (t4)  |:@02o3 v13‾1 L4
   149: (t4)  p2|:g+1^2g+g+f+1^2f+f+f+1^2f+f+f+8.^2.e16^1:|L8
   150: (t4)  p2|:|:f+.f+16r2:|f+.f+16r8 g+^1r8<|
   151: (t4)   g+16g+16g+16g+16r8g+.g+16r4>:|
   152: (t4)  <g+16r8g+16r8g+4r4>
   153: (t4)    |:|:f+.f+16r2:|f+.f+16r8|g+^1r8<
   154: (t4)   g+16g+16g+16g+16r8g+.g+16r4>:|
   155: (t4)  g+^1^4g+r8g+.g.f+^2.^8.e16^2.^8.f+16^1L16
   156: (t4)  <r16p1f+r16p2f+r16p1f+r16p2f+r16r4..>
   157: (t4)  p2L1ag+f+2.^8.f+16^4..f+16^2 ag+f+f+
   158: (t4)  ag+f+4^8.f+16^4^8.f+16^4^8.f+16^2
   159: (t4)  ag+f+ L12rf+f+f+f+f+f+r4rr:|
   160: (t4)  |:@03p2o4 f1^4f2f8.e16^2.^8.e16^1
   161: (t4)  f2.f8.f16^2f4..e16^2.^8.e16^1
   162: (t4)  f1^2f4f8.e16^2^8e4.^1
   163: (t4)  f1^2f4f8.e16^2&_r8‾<e4.^2e2:| @02o3
   164: (t4)  L1p2|:13a g+f+2... f+2f+2^16:|
   165:
   166: /---------------
   167: / MAIN
   168:
   169: (t5)  @03@k01 v06o4L16 q8
   170: (t5)  p1|:e1^4^16@02<p1erer4>
   171: (t5)  @03p1f+4g1^4@02<p1ggrgr4>@03p1|f4:|f4 r
   172: (t5)  @04@k05 v13‾2o0
   173: (t5)  p3arbr<drf+rar<c+rdrd+e1
   174: (t5)  r1
   175: (t5)  L8p3|:|:e2.c+.>g+16^2r8f+g+ag+2.f+.f+16^2.r4
   176: (t5)  <f+2.d.>b16^2.r8g+16a16a+8.^2.b16^2.r4:|
   177: (t5)  p3<|:|:f+.e16r2:|a.g+16r8e^1^2dc+>b16<c+.
   178: (t5)  |:f+.e16r2:|a.g+16r8|e^2.^8.<c+16^1:|
   179: (t5)  c+1>b16a16g+2 a.g+.f+^2.^8.g+16^2.e.f+16^1^2
   180: (t5)  p3a.g+.f+16e16f+2r8a4bg+2f+4e4f+1^2
   181: (t5)    a.g+.f+16e16f+2r8g+4ab2 b*2<c+*46e4f+1^1
   182: (t5)  |:7a8g+16f+16:|a16b8.ag+16f+16^2.^2^8.e.c+>
```


▶響子in CGわ～るどの絵本，そろそろ出てくれてもいいと思いますが。CGそのものも素敵ですが，文章がいいと思います。なにか忘れたものを，ふと感じさせるような……。
中村　学(21)福岡県

```
 183: (t5)  b4<c+>a^4f+ab2ag+4f+16e16f+1r1:|
 184: (t5)  p3e2.d8.e16^4>a4<d4g8.e16^2.d8.e16^2.r4
 185: (t5)  e2.d8.e16^2>a4<g8.e16^2.^8d16c16d4.c16>b16<c4>b16a
8.<
 186: (t5)  e2.^8d16>a16<e4.d16>a16<d4g8.e16^1g16a16g8e8d16e16
^4r4
 187: (t5)  |d16e8.^2>a8.<d16e16d16e4>a8<d4g8.e16^1^1
 188: (t5)  >g4.f4g8a8b8^2<d4>a8.g16^4..g*1a*11^1^8b8<c8d8
 189: (t5)  e2^8>a8<d8e8^4.d16>a16<d4g8.e16^1^2.a4
 190: (t5)  <e2.d8>a8<e4.d16>a16<d4g8.e16^1^2.r4
 191: (t5)  e8d16>a16^2<e8d16>a16^4.<e16d16>a8<d8g8.e16^1^2
 192: (t5)  g4a8e8 b1g+2e2f+1^2>
 193: (t5)  p3|:3a.g+ .f+16e16f+2r8a4bg+2f+4e4f+1^2
 194: (t5)  a.g+.f+16e16f+2r8g+4ab2 b*2<c+*46e4f+1^1
 195: (t5)  |:7a8g+16f+16:|a16b8.ag+16f+16^2.^2^8.e.c+>
 196: (t5)  b4<c+>a^4f+ab2ag+4f+16e16f+1r2:|r2
 197:
 198:
 199: (t6)  @03@k01 v06o3L16 q8
 200: (t6)  p1|:b1^4^16@02<p3brbr4>
 201: (t6)  @03p1b4<c1^4@02<p3ccrcr4>@03p3|c4:|c4 r
 202: (t6)  @02@k03 v13_1o2
 203: (t6)  p3arbr<drf+rar<c+rdrd+e1
 204: (t6)  r1
 205: (t6)  L8p3|:|:e2.c+.>g+16^2r8f+g+ag+2.f+.f+16^2.r4
 206: (t6)  <f+2.d.>b16^2.r8g+16a16a+8.^2.b16^2.r4:|
 207: (t6)  p3<|:|:f+.e16r2:|a.g+16r8e^1^2dc+>b16<c+.
 208: (t6)  |:f+.e16r2:|a.g+16r8|e^2.^8.<c+16^1:|
 209: (t6)  c+1>b16a16g+2 a.g+.f+^2.^8.g+16^2.e.f+16^1^2
 210: (t6)  p3a.g+.f+16e16f+2r8a4bg+2f+4e4f+1^2
 211: (t6)    a.g+.f+16e16f+2r8g+4ab2 b*2<c+*46e4f+1^1
 212: (t6)  |:7a8g+16f+16:|a16b8.ag+16f+16^2.^2^8.e.c+>
 213: (t6)  b4<c+>a^4f+ab2ag+4f+16e16f+1r1:|
 214: (t6)  p3e2.d8.e16^4>a4<d4g8.e16^2.d8.e16^2.r4
 215: (t6)  e2.d8.e16^2>a4<g8.e16^2.^8d16c16d4.c16>b16<c4>b16a
8.<
 216: (t6)  e2.^8d16>a16<e4.d16>a16<d4g8.e16^1g16a16g8e8d16e16
^4r4
 217: (t6)  |d16e8.^2>a8.<d16e16d16e4>a8<d4g8.e16^1^1
 218: (t6)  >g4.f4g8a8b8^2<d4>a8.g16^4..g*1a*11^1^8b8<c8d8
 219: (t6)  e2^8>a8<d8e8^4.d16>a16<d4g8.e16^1^2.a4
 220: (t6)  <e2.d8>a8<e4.d16>a16<d4g8.e16^1^2.r4
 221: (t6)  e8d16>a16^2<e8d16>a16^4.<e16d16>a8<d8g8.e16^1^2
 222: (t6)  g4a8e8 b1g+2e2f+1^2>
 223: (t6)  p3|:3a.g+ .f+16e16f+2r8a4bg+2f+4e4f+1^2
 224: (t6)  a.g+.f+16e16f+2r8g+4ab2 b*2<c+*46e4f+1^1
 225: (t6)  |:7a8g+16f+16:|a16b8.ag+16f+16^2.^2^8.e.c+>
 226: (t6)  b4<c+>a^4f+ab2ag+4f+16e16f+1r2:|r2
 227:
 228:
 229: (t7)  @03@k01 v06o4L16 q8
 230: (t7)  p2|:c+1^4^16@02<p2c+rc+r4>
 231: (t7)  @03p2c+4e1^4@02<p2eerer4>@03p2|d4:|d4 r
 232: (t7)  @02@k02 v13_1o2
 233: (t7)  p3arbr<drf+rar<c+rdrd+e1
 234: (t7)  r1
 235: (t7)  L8p3|:|:e2.c+.>g+16^2r8f+g+ag+2.f+.f+16^2.r4
 236: (t7)  <f+2.d.>b16^2.r8g+16a16a+8.^2.b16^2.r4:|
 237: (t7)  p3<|:|:f+.e16r2:|a.g+16r8e^1^2dc+>b16<c+.
 238: (t7)  |:f+.e16r2:|a.g+16r8|e^2.^8.<c+16^1:|
 239: (t7)  c+1>b16a16g+2 a.g+.f+^2.^8.g+16^2.e.f+16^1^2
 240: (t7)  p3a.g+.f+16e16f+2r8a4bg+2f+4e4f+1^2
 241: (t7)    a.g+.f+16e16f+2r8g+4ab2 b*2<c+*46e4f+1^1
 242: (t7)  |:7a8g+16f+16:|a16b8.ag+16f+16^2.^2^8.e.c+>
 243: (t7)  b4<c+>a^4f+ab2ag+4f+16e16f+1r1:|
 244: (t7)  p3e2.d8.e16^4>a4<d4g8.e16^2.d8.e16^2.r4
 245: (t7)  e2.d8.e16^2>a4<g8.e16^2.^8d16c16d4.c16>b16<c4>b16a
8.<
 246: (t7)  e2.^8d16>a16<e4.d16>a16<d4g8.e16^1g16a16g8e8d16e16
^4r4
 247: (t7)  |d16e8.^2>a8.<d16e16d16e4>a8<d4g8.e16^1^1
 248: (t7)  >g4.f4g8a8b8^2<d4>a8.g16^4..g*1a*11^1^8b8<c8d8
 249: (t7)  e2^8>a8<d8e8^4.d16>a16<d4g8.e16^1^2.a4
 250: (t7)  <e2.d8>a8<e4.d16>a16<d4g8.e16^1^2.r4
 251: (t7)  e8d16>a16^2<e8d16>a16^4.<e16d16>a8<d8g8.e16^1^2
 252: (t7)  g4a8e8 b1g+2e2f+1^2>
 253: (t7)  p3|:3a.g+ .f+16e16f+2r8a4bg+2f+4e4f+1^2
 254: (t7)  a.g+.f+16e16f+2r8g+4ab2 b*2<c+*46e4f+1^1
 255: (t7)  |:7a8g+16f+16:|a16b8.ag+16f+16^2.^2^8.e.c+>
 256: (t7)  b4<c+>a^4f+ab2ag+4f+16e16f+1r2:|r2
 257:
 258: /-----------------
 259: / BASS
 260:
 261: (t8)  @01p3@k02v14~1o2L16 q8
 262: (t8)  |:a8.<ee8.>aa8.<ee8.>aa8.<ee>arar4<e4>
 263: (t8)  f8.<cc8.>ff8.<cc8.>ff8.<cc>frfr4<c4>:|
 264: (t8)  |:7re:|re<e1r1>
 265: (t8)  |:|:|:4a8.<e|e8.>a:|<e8>a8|:4f+8.<c+|c+8.>f+:|<c+8
>f+8
 266: (t8)  |:4b8.<f+|f+8.>b:|<f+8>b8|:4e8.b|b8.e:|b8<e8:|L8
 267: (t8)  |:|:d.d16r2:|d. d16r8 >a1g+16a16r8<|
 268: (t8)  >a16a16a16a16r8g+.a16r4:| a16r8a16r8a4r4<
 269: (t8)    |:d.d16r2:|d. d16r8 >a1g+16a16r8
 270: (t8)  <a16a16a16a16r8g+.a16r4
 271: (t8)    |:d.d16r2:|d.<d16r8>>a^1^4ar8
 272: (t8)  a.g+.f+^2.f+8.e16^2.e8.f+16^2<f+2L16
 273: (t8)  r16p1f+r16p2f+r16p1f+r16p2f+r2>L8p3
 274: (t8)  |:4d.a.<d16r16d.>a16a<d16r16>e.b.<e16r16e.>
 275: (t8)  b16b<e16r16>f+.<c+.f+16r16f+.c+16c+f+16|
 276: (t8)  r16>f+.<c+.f+16r16f+.c+16c+f+16r16>:|
 277: (t8)  r16 L12rf+f+f+f+f+f+r4rrL16:|<
 278: (t8)  |:4|:d8.d^2.:|>|:e8.e^2.:|<:||:4|:d8.d^2.:||:e8.e^
2.:|:|
 279: (t8)  L8>|:13d.a.<d16r16d.>a16a<d16r16>e.b.<e16r16e.>
 280: (t8)  b16b<e16r16>f+.<c+.f+16r16f+.c+16c+f+16r16>
 281: (t8)  f+.<c+.f+16r16f+.c+16c+f+16r16>:|
 282:
 283: /-----------------
 284: / DRUMS
 285:
 286: (t9)  @01v09p3L16o2 q8
 287: (t9)  |:4c8.c<<d8>>c8 c8.c<<d8>>c8 c8.cc<<d8d4>>c<<d8>>c
8:|
 288: (t9)  c4.rcc4<<d8>>c<<d8^1>>rc8.c8.c8.<<dd8>>c8
 289: (t9)  |:L16|:32c8.cc8<<d>>c:|
 290: (t9)  |:4c8.c16r4<<d4>>c8.c16r4<<d4>>c8.c8.c8r4|<<d4r8>>
 291: (t9)  rrrrc16c8.c8.c8.c8.c8.<<d16d16>>:|
 292: (t9)  r4<<d4.>>c16c8.r8<<d4>>c8.c8.c4.
 293: (t9)  |:3<<d8.>>c8cc8<<d8>>c8|c4:|
 294: (t9)  <<d>>c8c8c8c^8.<<ffcd8>>
 295: (t9)  |:15c_18<<d8>>~c<<e8.>>_<<e>>~c8.c<<e8>>_<<ee>>~:|
 296: (t9)  L12c<<eeeeee>>rc12^4:|L16
 297: (t9)  |:16c4..c8.c8<<c8>>_24c~c c4..c8.c8<<f8c>>c:|
 298: (t9)  |:104c8.cc8<<d>>c:|
 299:
 300: /-----------------
 301: / H.HAT
 302:
 303: (t10) v08p3L16o6q8
 304: (t10) |:4|:5~9c_ccc:|c~d_c~d_c8c24c24c6cc:|
 305: (t10) r|:6~d_c:|c8.|:8cc~d_c:|
 306: (t10) |:|:16c8ccc8ccc24c24c24ccc8cc:|
 307: (t10) |:4|:~d8.d_ ~c_ccc ~c_ccc:|~d8.d_~c_c~d8_ rccc
 308: (t10) ~c_ccc c24c24c24cc |:5~c_ccc:|:|
 309: (t10) |:3rccc cccc cccc cccc:||:4c~d_:|cr4..
 310: (t10) |:60cc~d_c:|r1:|
 311: (t10) |:128cc~d_c:|
 312: (t10) |:52c8ccc8ccc24c24c24ccc8cc:|
 313:
 314: /-----------------
 315: / CYMBAL
 316:
 317: (t11) @01v09p3L1o6
 318: (t11) o4|:4r1r2.c4:|o6
 319: (t11) r*192br
 320: (t11) |:L1_18a~r*2880
 321: (t11) v04|:4r2>>a4r2a4<<r2.|>>a4r4.r1 a16a16<<:|v09
 322: (t11) r4r1q8b8.b8.b4^2._18aaar8~
 323: (t11) r*2880
 324: (t11) L12q8r r2rb12^4&:|L1
 325: (t11) >|:128c4:|
 326: (t11) r*9984
 327:
 328:
 329: (p)
```

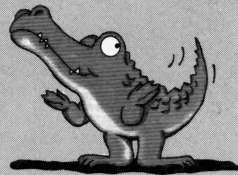
リスト5　PASSING BREEZEの音色コンフィグファイル

```
1=clapm1.pcm,p-1,v5
2=bass1.pcm,v30
3=cow808.pcm,v11,p3
4=casta.pcm,p-11,v17

.o2c=sph_bd.pcm,v164,m2
.o2d=.o2c,v112

.o4c=sph_lt.pcm,v120
.o4d=sph_sd.pcm,v151
.o4e=.o4d,mo2c
.o4f=sph_mt.pcm,v210
.o4a=sph_ht.pcm,v119
.o4b=tablna.pcm,v15,p-9,m3

.o5c=side1.pcm,v37,p-2,m4
.o5d=clapm1.pcm,p-1,m1,d930,v41
.o5b=shakerm1.pcm,p2,v140

.o6c=sph_ch.pcm,v82
.o6d=sph_oh.pcm,v82
.o6a=sph_cc.pcm,v188
.o6b=sph_cc.pcm,v200

.erase 1
.erase 2
.erase 3
.erase 4
```

リスト6　PASSING BREEZEのカウンタ表示

```
 1:00009540 00000000   2:00009540 00000000   3:00009540 00000000   4:00009540 00000000
 5:00009540 00000000   6:00009540 00000000   7:00009540 00000000   8:00009540 00000000
 9:00009540 00000000  10:00009540 00000000  11:00009540 00000000
```

▶OS-9/X68030に興味をもってしまった。やっぱり，あのマルチユーザー，マルチタスクの威力はすごいと思う。
高木　俊之(19)新潟県

(善)のゲームミュージックでバビンチョ

西川善司

ゲームミュージックライブ電撃'93

毎年恒例のゲームミュージック・フェスティバルだが，今年は主催者側の都合で，ほとんど同じような内容のイベントが，ほぼ同時期に2カ所で開催された。1つは7/30〜8/1にクラブチッタ川崎で行われた「JAPAN GAME MUSIC LIVE 1993」，そして8/19〜8/20の日本青年館(千駄ヶ谷)での「ゲームミュージックライブ電撃'93」だ。日にちの選択肢が増えていいのかもしれないけれど，客足が分散してしまった感じも……。私は後者のほうに行ってきたので簡単にレポートをしよう。

まず，初日。筆頭バンドはNECアベニューの葉山宏治&ブラザーズ。ボディビルダー乱入，拡声器を使っての「超兄貴」ネタ連発でファンを楽しませていた。私はどーも馴染めず(私はご家庭ゲームにうとい)，「もぅ……だめだー」を連発する葉山宏治に対して，私のほうがもうダメになりそうだった。あの世界に没頭するには，私自身の精進が必要のようだ。

そんで次はデータイーストのゲーマデリック。軽快なゲーマデリックのテーマと共にメンバーの登場，そして和製ラップのMr.Kの乱入でリズミカルに幕を開けた。「カルノフ」や「スーパーコップ」と日本語ラップ曲，「ファイターズヒストリー・メドレー」や「空牙」などの正統派ゲームミュージックを快演，あげくの果てには新作ゲーム「フラワーバスターズ」の曲までも披露し，実に賑やかなステージとなった。しかしインスト演奏中にやることのなくなった日本語ラッパーMr.Kの立場は複雑だろう。もう踊るしかないってやつか。踊ってたけど。

最後はコナミの矩形波倶楽部。「Pop'nツインビー」のマドカの声を担当した國府田マリ子も参加。思ったよりかわいくて音も採れているのでビックリ。國府田マリ子の「Pop'nツインビー」のボーカル曲で始まり，「グラディウスIII」，「Pop'nツインビー・エンディング」とメロディアスなインスト・フュージョンを続けて演奏。リーダー古川氏のギターはうまいし，2人のキーボードも絶品，さらにコンピュータシーケンスのバックで音が厚い，さすがトリを務めるだけはある。ところで國府田マリ子，いまどき珍しいブリッコ系だ。途中インスト演奏の際に用済みとなり，舞台から下がるときに「國府田はこれでさよならです，ぐすん」，あまりの歯がゆさに私は前の座席のカバーを握り破ってしまった。恐るべしマドカ。

そんでもって2日め。まず1番目はいまだバンド名の謎が解明されないファルコムのJ.D.K.BAND。ジャマイカ・ダンシング・キッズという説が有力だが定かではない。見てるだけで暑苦しくなる長髪集団が威勢よく飛び込んできて，「イース」の曲で幕を開けた。ちょっとノリのよくない観客に「おい，静かになるなよ」と喝を入れていたが，観客は全員「静かにさせるなよ」と突っ込んでいたに違いない。しかし，ギターもうまいしボーカルもそこそこだし，なぜ人気がいまいちなのかは私にもわからない。誰か教えてくれ。

さて，お次はメンバーが全員女性のカプコンのアルフ・ライラ。サーカスの猛獣使いみたいな怪しげな衣装で登場。演奏曲はもうこれしかないんじゃないかという噂もあるが「ストリートファイターII」。ギタリストがバラードでいきなりバイオリンを弾いてしまうなどの意表をつくパフォーマンスもあり，観衆のどよめきも。ただ，ボーカルのシャウトがものすごく甲高くハウリングみたいな音にもなったりして，私は唾を飲み込んで息を止めて耳が痛いのをこらえていた。あやうくジャイアンリサイタルになりかけていたぞ。

締めはGM界の大御所セガのS.S.T.BANDだ。「バーチャレーシング」の新曲で颯爽と登場，館内あっというまの歓声の嵐。

スゴイ人気。この新曲「POLYPHONIC CONTINENT」は単調なシーケンスにメンバー全員がアドリブ・タッチで順番に演奏に参加してくるタイプの曲。ちょうどアルバム「BLIND SPOT」の「SEVENTH FLIGHT」のようなタイプだ。KX-5をかついでの超高速シンセソロは圧巻だった。続いて「バーチャレーシング」からまたまた新曲「STREAM」。こちらはロック系のスピード感溢れるサウンドで「アフターバーナー」以来のヒットの予感がする。そのあと，メンバーがサルやトトロの格好をして日本語ラップに挑戦。これははっきりいってつまらなかった。そのほか「バーミリオン」「ファンタシースター」「アフターバーナー」などの定番を演奏し，幕を閉じた。さすがトリを務めるバンドだけあって実力も人気もあり，大御所の名に相応しい舞台を見せてくれた。日本語ラップはほとんど志の輔の「ペヤング・ソース焼きそばぁっ」のノリだったのでいらないが，「バーチャレーシング」のほうは早くCDが欲しいといったところ。

というわけで，ライブレポートを終わる。なんか矩形波倶楽部とS.S.T.BANDは今回のイベントをもって現メンバーでの活動が終わるとか。今後どうなるのだろうね。

●スティールガンナー
VHS:VIVL-102　5,800円(税込)
ビクター・エンターテインメント　発売中

ナムコのガンシューティングゲーム「スティールガンナー」と「スティールガンナー２」の映像とサウンドを収録したビデオ。攻略ビデオというよりはゲーム映像を楽しむといった趣向の１作だ。

いわゆるポリゴンものではなくスプライトものなので緻密な３次元感覚はあまりないものの，スピード感ある演出と迫力のサウンドは見る者に圧倒的な臨場感を感じさせる。

お勧め度　7

●スターフォックス
CD:TECD-25275　2,500円(税込)
テイチク　9/22発売

任天堂が総力を結集して制作した３Dポリゴン・シューティング「スターフォックス」のサントラ・アルバム。アレンジバージョンは残念ながらおチープな音でがっかり，アンビエンス系のエフェクトは，売り物なんだからもっとしっかりするべき。子供だましはいかん。さらに宇宙の香りがするあの広大なメインテーマがへぼいアレンジでだいなし。

さてオリジナルバージョンは全曲，隠れ面までを網羅して収録。こちらは２ループ収録で１曲を堪能できる。メインテーマである「L8b-4.<ff2 r4efgcde g4.f16e16f2 r4 f4g4a4 b-4.e-16d16e-4b-4 b-4.e-16d16e-4b-4 <c1> f1」から発想を膨らませたと思われる曲がたくさんあって世界の統一感があっていい。映画音楽みたいだ。

お勧め度　9

おまけ

●GREAT WALL
CD:TTRC-0002　2,000円(税込)
TROUBADOUR RECORD

友人が「買い過ぎちゃったので善ちゃん買ってーん」と電話してきたので「まぁ，いいか，買っちゃろう」ということで手に入れたのがこのCDだ。

ゲームミュージック界の「大御所」といわれるコンポーザが集まって作り上げたというこのCD，期待度満点ではないか。「ベラボーマン」の中潟憲雄，「ドラゴンセイバー」のメガテン細江，「ギャラクシアン」の相原隆行や「ローリングサンダー２」の佐宗綾子，「伝説のオウガバトル」の崎本仁，あの古代祐三も参加しているではないか。しかし以前，同じ友人から同じような「著名GMミュージシャンによるCD！」というやつを聴かせてもらったことがあったが，それはなんだかジャイアンの親戚かとも思えるオンチなボーカル曲や，「一足お先に，パヤパヤ」とか死語に近い歌詞を気持ちよさそうに歌う脳天気女性ボーカル曲が入っていてガッカリさせられたことがあった。そこで今回のCDでも私は懐疑的先入観で聴かざるを得なかったのだが，いやいやどっこい，そういうひねくれた気持ちをすっぱり払い去ってくれるほどの出来映えであった。極上テクノサウンドの集大成。変にサンプリングネタに走らない「曲を聴かせてくれる」「テクノ」といった表現がいちばん適切だろうか。このCDの入手経路を知っている方がいたらここで紹介したいので連絡待つ。

終わりに

●８月某日

埼京線に乗っていたときの出来事。隣の座席の２人のガキどもが，停車する駅名を順番に駄洒落る遊びを大きな声でやっていた。私は眠かったので目を閉じていたが，大きな声でやってるために，どうしても耳に飛び込んでしまう。「川越の川怖えぇ～」「与野は広いよの～」「北戸田にいま来たと(こ)だ」私は怒りにうち震えていたが，板橋に停車したときに小僧Aが「痛っ！　バシ！(殴るポーズつき)」と叫び，それに妙にウケてしまった私は，目を閉じたまま思い切り吹き出してしまった。そして，その後も「痛っ！　バシ！」が耳について離れず，終点の新宿まで目を閉じたまま笑いをこらえる変なお兄さんになってしまったのであった。

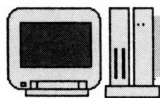
●８月某日

八王子駅のバス停でバスを待っていたところ，サマーランドへ行くと思われる元気のいい小僧３人が私の後ろに並んだ。そこで小僧Aが突然「あぁっ！」と叫び百貨店そごうのビルの上を指さし，私はとっさにそちらのほうに目を向けた。しかしその方向には何もなく，その焦点よりも遥か近辺の視界の下で「ばっかが見るー」と小僧Aがさらに叫び，２人の小僧の悔しがっている姿が目に映った。

●８月某日

先日高校の同窓会があった。野瀬という奴がＪリーグの「♪オレーオレオレオレー」のテーマをもじった「♪野瀬一野瀬野瀬野瀬一」に乗せられて酒を多量に飲まされ，意識が朦朧としていた。重たい彼の処分に困った我々は，段ボールの切れ端に「国分寺で降ろしてください。怪しい者ではありません」と書いた札を彼の首から下げて，中央線に乗せ捨ててしまった。みんなもこうならないように酒はほどほどに，そして友達は選べ。ではまた来月。

スターフォックス

GREAT WALL

MS-DOSマシンとの接続

電机本舗　由井　清人　Yui Kiyoto

X68000といろいろなコンピュータをつないでファイル転送を行おうというこの連載。今回はX68000とMS-DOSマシン一般のあいだでファイルをやり取りするためのプログラムを作成します。基本部分は1993年7月号を参照してください。

前回は，X68000同士での転送を考えてみました。今回はX68000と異機種での接続を考えてみます。

さて，その前に前回の記事中誤りがあったので明記しておきます。通信設定を行うSPEED.Xの実行画面に若干の誤りがありました。正しくは，画面1となります。

前回作った転送プログラムはXon/Xoffを使用しますと動きません。オフにしてください。ややこしいのですが，ishとLHaを使ったバッチファイルのときには，オンにしておくとよいでしょう。コマンドラインからの設定パラメータをあげておきます。バッチファイルの中に組み込んで使用してください。

●転送プログラムのとき
　SPEED 9600 b8 pn s1 none
●ishとLHaのとき
　SPEED 9600 b8 pn s1 xon

PC-9801（汎MS-DOS）との接続

まず，ここではPC-9801との接続プログラムを作ってみましょう。基本的には前回の転送プログラムをそのまま使用します。前回の転送プログラムはX68000特有の機種依存コードを7月号リスト5“IO.H”に閉じ込めてありましたので，この部分を変更しました。転送プログラムは通信機能の1文字入力と1文字出力だけを利用してファイルを転送します。ですから，X68000上のこの機能をMS-DOSのものと置き換えてあげればよいことになります。

MS-DOSのAUX（補助入出力-RS-232C-）への1文字の入出力がありますので置き換えました。具体的には，7月号リスト3とリスト4の数行を変更します。MS-DOSの機能のみを使用していますので，FMRやTOWNSなどでも動くはずです。

●7月号リスト3の変更情報
　70：#include　”IO.H”　　(旧)
　70：#include　”MSDOS.H”(新)
　71：　　　　　　　　　　(旧－空白行)
　71：#include　”XCLS.H”　(新)
●7月号リスト4の変更情報
　69：#include　”IO.H”　　(旧)
　69：#include　”MSDOS.H”(新)

そして，今回作った新しいファイル“MSDOS.H”と“XCLS.H”をエディタより入力してください。ファイルの内容はリスト1および2に示します。

コンパイル&使用方法

MS-DOS側ではQuick-Cのver.2を使用してコンパイルしました。コンパイルオプションは規定値です。

PC-9801で使用するにあたり，ハードウェアの設定とDOSの設定が必要です。

a)　ハードウェアの設定
　ディップスイッチ1の5と6をオフにしてください。
b)　DOSの設定

IBM PC属亜科における問題点

この系統のマシンは頭痛の種です。基本的にはIBM　PCのクローンです。PC/AT,AX,J-3100,PS/2,PS/55そしてDOS/Vマシンと呼ばれるものです。

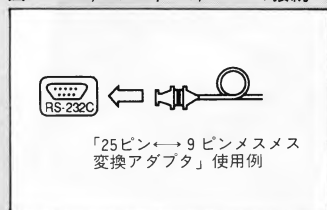
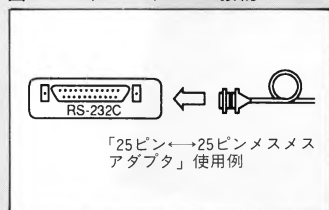
このタイプの問題は，通信コネクタの形状が2系統あること。そして，共に通常のRS-232Cコネクタの形状がパソコン側がメス型であるのに対しオス型であるということです。

これらに対処するために，パソコンショップなどで極性変換器（ジェンダーチェンジャーと呼ばれている）が販売されています。これらの機器を利用するのがよいでしょう。

図1　AX, J3100, PC/ATへの接続

「25ピン←→9ピンメスメス変換アダプタ」使用例

図2　PS/55 PS/52への接続

「25ピン←→25ピンメスメスアダプタ」使用例

画面1　SPEED.Xの設定

MS-DOSに通信制御ドライバを組み込んでおく必要があります。詳しくはMS-DOSのマニュアルを読んでください。通常は，環境設定ファイル“CONFIG.SYS”へ次の行を加えます。

DEVICE=DOS¥RSDRV.SYS

RSDRV.SYSは通信制御のデバイスドライバです。どのサブディレクトリに格納されているかはユーザーにより異なります。

c)　通信速度の設定

X68000同様に通信速度の設定をSPEEDコマンドで行います。ただし若干パラメータが異なるので注意してください。参考までに9600bpsでの設定例を次に示します。バッチファイルなどを組んでおくとよいでしょう。

SPEED r0 9600 b8 pn s1 none

プログラムの実行方法はX68000版と同一です。具体的には，受信側で，

A>TENRSI

と実行しておき，次いで送信側で，

A>TENRSO A¥AUTOEXEC.BAT

のように実行してください。

終了方法は送信側で，

A>TENRSO -E

と実行してください。受信側がこれで終了します。

IBM PC,AX,PS/55,J-3100,DOS/Vとの接続

次に考えるのはIBM PC系のマシンとの接続ですが，これらのタイプのマシンは実はDOSレベルで通信機能を持っていません。

厳密には，持っているのですが，正常に動かないのが現状です。たとえば，通信をしている最中にディスクへのアクセスが発生するとCPUは通信処理を放り出し，ディスクの読み書きを始めます。この間に送られてきたデータはこぼれます。

このようなわけで，通信制御はハードウェアを直接制御し，ディスクアクセスと通信制御で上手にタイミング調整してあげないとうまい具合にいきません。

PC-9801のとき同様，“IO.H”用差し替えファイル“IBM.H”(リスト3)を用意しておきました。これを入力してください。

また，7月号のリスト3，4の修正情報を示します。

●7月号リスト3の変更情報

70：	#include　"IO.H"	(旧)
70：	#include　"IBM.H"	(新)
71：		(旧－空白行)
71：	#include　"XCLS.H"	(新)
81：		(旧－空白行)
81：	_ibm_inz()；	(新)

●7月号リスト4の変更情報

69：	#include　"IO.H"	(旧)
69：	#include　"IBM.H"	(新)
82：		(旧－空白行)
82：	_ibm_inz()；	(新)

DOS上でコンパイルすれば使用できます。

リスト1　MSDOS. H

```
 1: /* MS-DOS 補助入力ファンクションNo */
 2: #define         AUX_IN  0x03
 3: #define         _AUXOUT 0x04
 4:
 5:
 6:
 7:
 8: #include        <stdlib.h>
 9: #include        <dos.h>
10: #include        <memory.h>
11: #include        <sys¥types.h>
12: #include        <sys¥stat.h>
13: #define         _RS_OUT(a)      bdos( _AUXOUT, a )
14: #define         DLYTIME(a)      dlytime(a)
15:
16:
17:
18:
19: /*****************************************************************************
20: *       _rs_in_x RS232Cより一文字受信す
21: *       in      :       nothing
22: *       return  :       受信キャラクタ
23: *****************************************************************************/
24: unsigned char   _rs_in_x()
25: {
26:         return( bdos(AUX_IN) );
27: }
28:
29:
30: /*****************************************************************************
31: *       dlytime 時間遅延
32: *       return  :       non
33: *****************************************************************************/
34: dlytime(lp)
35: int lp;
36: {
37:   int   a;
38:   int   b;
39:   int   c;
40:   int   i;
41:
42:         for( i=0 ; i< lp ; i++ ) {
43:                 a = 123;
44:                 b = 111;
45:                 c = a*b;
46:         }
47: }
```

リスト2　XCLS. H

```
 1:
 2: /*****************************************************************************
 3: *       _x_close        ファイルクローズ
 4: *       in      : f     ファイルハンドル
 5: *       return  :       status  0=NORMAL, 0!=ERR
 6: *****************************************************************************/
 7: int     _x_close( f )
 8: int     f;
 9: {
10:         write( f, __buf, (unsigned int)_buf_ptr - (unsigned int)__buf);        /* flush
    */
11:         DLYTIME( 1000 );
12:
13:         _dos_setftime( f, dt, tm );
14:         DLYTIME( 1000 );
15:
16:         return( close( f ) );
17: }
```

PC-98属に関する混迷

PC-98系は通信機能がBIOS(X68000でいうIOCSのこと)できちんと対応されているので問題はありません。少なくとも，間違いなく動くという意味においてはです。

ただ，いかんせん，設計が古くまたハード設計に機種ごとの互換性がありません。実は，98にはCPUクロックに8MHz系と5MHz系の２つの系統があり混乱しています。

通信というのはこのクロックをストップウォッチのように計り，一定期間ごとに通信データをやりとりします。そのため，通信速度にはクロックの速さが大きな影響を与えます。

具体的にはクロックは分周されLSIに供給されます。ですから，このクロックのタイミングで割り切れる速度しか選択できません。

その結果，8MHz系で9,600bps，5MHz系で153,600pbsという差がでてしまいます。

そして，困ったことにNEC系のノートはCPUに5MHz系を供給していても通信LSIには別に8MHz系を供給するということをしています。このあたり，エプソンのノートは両方選べるなど工夫があり，ことに通信に関しては一日の長があります。

PC-9801同様に変更部分をリストにしておきます。

このタイプのマシンはDOSの管理下よりRS-232Cを実用的レベルで使用できません。

リストでは通信LSIを直接制御するようにしています。このために，DOS上での設定は不要です。今回の転送プログラムが強引に初期化します。

プログラムの実行方法はX68000版と同じです。

リスト3　IBM. H

```
 1:
 2: #include        <stdlib.h>
 3: #include        <dos.h>
 4: #include        <memory.h>
 5: #include        <sys¥types.h>
 6: #include        <sys¥stat.h>
 7: #define         _RS_OUT(a)      _rs_outibm( a )
 8: #define         DLYTIME(a)      dlytime(a)
 9: #define COM_PORT        0x3F8           /* COM1=3F8,COM2=2F8,COM3=3E8,COM4=2E8  */
10: #define IOP_IER         1               /* interrupt enable register            */
11: #define LCR_DLAB_BIT    0x80            /* divisor addres latch bit             */
12: #define IOP_TX_DATA     0
13: #define IOP_RX_DATA     0
14: #define LCR_BITS        3               /* 8 data, no parity, no break          */
15: #define IOP_LCR         3               /* line control register                */
16: #define IOP_LSR         5               /* line status register                 */
17: #define LSR_RX_READY    1
18: #define LSR_TX_READY    32
19: #define BAUD_9600       7
20:
21: /*unsigned char   _rs_outibm(); */
22:
23:
24:
25: /**************************************************************************
26: *       _rs_in_x RS232Cより一文字受信す
27: *       in      :       nothing
28: *       return  :       受信キャラクタ
29: **************************************************************************/
30: unsigned char   _rs_in_x()
31: {
32:         while( (inp(COM_PORT+IOP_LSR)&LSR_RX_READY)==0 );
33:         return( inp( COM_PORT+IOP_RX_DATA ) );
34: }
35:
36:
37:
38: _ibm_inz()
39: {
40:         outp( COM_PORT+IOP_IER, 0 );
41:         outp( COM_PORT+IOP_IER+IOP_LCR, LCR_BITS );
42:         _ibm_bps( BAUD_9600 );
43: }
44:
45: _ibm_bps( bps )
46: unsigned int    bps;
47: {
48:   unsigned int  clk;
49:   unsigned char clk1;
50:   unsigned char b;
51:
52:         clk = 1536;             /* クロックベースset            */
53:         clk >>= bps;            /* クロック設定                  */
54:         clk1 = (unsigned char)( clk >> 8 );
55:
56:         b = inp( IOP_LCR + COM_PORT );
57:         outp( IOP_LCR + COM_PORT, (unsigned char)(b | LCR_DLAB_BIT) );
58:
59:         outp( COM_PORT, (unsigned char)clk );
60:         outp( COM_PORT + 1, clk1 );
61:
62:         outp( COM_PORT + IOP_LCR, (unsigned char)b );
63: }
64:
65: _rs_outibm( c )
66: unsigned char   c;
67: {
68:         while( (inp(COM_PORT+IOP_LSR) & LSR_TX_READY)==0 );
69:         outp( COM_PORT+IOP_TX_DATA, c );
70: }
71:
72: /**************************************************************************
73: *       dlytime 時間遅延
74: *       return  :       non
75: **************************************************************************/
76: dlytime(lp)
77: int lp;
78: {
79:   int   a;
80:   int   b;
81:   int   c;
82:   int   i;
83:
84:         for( i=0 ; i< lp ; i++ ) {
85:                 a = 123;
86:                 b = 111;
87:                 c = a*b;
88:         }
89: }
```

通信制御の基礎メカニズム

いままで，ざっと話を進めてきて，SPEED.XやHuman68kの機能を組み合わせれば通信によるファイル転送ができあいで実現することがわかったと思います。

問題となるのは，通信ケーブルの結線，そしてSPEED.Xによる設定方法くらいです。しかし，これとても，たくさん選択肢があるというだけで，実際には定型でしか使いません。基本的にはブラックボックス化されていることがらなのです。

さて，ここでは，このブラックボックスを覗いてみようと思います。これから使う，もしくは普段使っているRS-232C通信の内側がわかりパソコン通信などの際に柔軟に対処できるようになるでしょう。

次回はモデムを使ったファイル転送を考えているので予備知識としてぜひ押さえておいてください。

調歩同期式通信というもの

通常使う通信方式は調歩同期式（非同期式ともいう）と呼ばれるものです。

この方式は，1データ（通常は1バイト）単位でデータを送ります。送り方は次のようなものです。データは，スタートビット，ストップビットというパルスに挟みこんで送るかたちになります（図1）。

思い出してください。SPEED.Xの画面設定にあったストップビット，スタートビットの指定とはこのパルスを出力する時間単位の指定するものだったのです。

データ長とパリティ

このようにしてデータを送るのですが，一方的にタレ流すため伝送エラーによりビットが潰れる可能性があります。そこで，通常8ビットのデータのうち1ビットをエラーチェック用として確保しておき，残り7ビットをデータ転送に使う考え方が出てきました。

7ビットのとき，1ビットクリップされますからバイナリや漢字データは伝送できません。7ビットでのみ表現できる英数字を前提にしたアメリカンカルチャーです。

それでは不便なのでデータ８ビット＋エラービットという形式も現れましたが，あまり使用されてはいません。SPEED.Xではデータ長５～８ビットの範囲で，それぞれにエラービットを加えるかどうか指定できます。

そしてエラーチェックは奇数式と偶数式があります。

フロー制御というもの

最後に，SPEED.Xで指定しているXonについて説明しておきます。これはフロー制御と呼ばれるものです。

通信制御というのは，コンピュータの実に苦手とする分野なのです。

ノイマン（逐次処理）型の現在のコンピュータはプログラムを１行ずつ実行していきます。

普段は，たったひとつのCPUがひとりで動いています。ですから速度が遅くて（ユーザーが）イライラする

ことはあっても処理が狂うということはありません。

しかし，通信制御ですとそうはいかないのです。2台のパソコンがそれぞれ送信と受信を行います。もし，両者で処理がズれると処理は止まります。

フロー制御というのは両者の調停をする処理です。

データ転送をしているときに受信側の処理速度が「常に」送信側を上回っていれば問題ありませんが，しばしば，遅れます。

そして，その瞬間，データの「とりこぼし」が発生します。このような事態を回避するためにフロー制御が考えられました。受信側が遅れがちになると，「待った」を送信側にかけるもの。それがフロー制御です。

Xon/Xoffと呼ばれる方式

フロー制御はソフトウェア式とハードウェア式の2つがあります。

ソフトウェア式は，送信側に「待った」をかけるのにXoffと呼ばれる文字コード（1バイト）を送ります。

送信側はこれを受けると，送信を一時停止します。そして，受信側から「もういいよ！」と転送再開を知らせるXonが送られてくるまで待ち続けます。

SPEED.XでのXonはこれの使用を意味します。

この方式は，ASCIIデータ転送においてのみ有効ということに注意してください。バイナリデータを転送し，もし，このなかにXoffと同じ値のデータが入っていると転送はロックしてしまいます。

よくパソコン通信などで，XMODEMプロトコルでXon/Xoffフローがおかしくなるというのはこのあたりに由来します。

プログラムがXon/Xoffを制御すればあまり問題にならないのでしょうが，X68000(PC-9801，Macintoshもそうです)は，IOCSなどの内部で処理しますから，データでXoffが送られてくると，途中でシステムが，これはデータではなく制御情報であると誤って認識し，送信をロックしてXoffを捨ててしまいます。

ハードウェアフロー

このような欠点を克服したものとしてハードウェアフロー方式があります。

この方式は，「待った」をかけるのにRS-232Cの通信制御ピンのRTSとCTSを使用する方式です。

受信側がCTS信号をオンにすることにより送信中断を相手に知らせる方式です。

この方式は，当然，ケーブルがRTS,CTSをきちんと接続されているときに使用できます。前回，掲載したケーブルの結線図はこれをきちんと結んでいます。

近況報告&次回予告？

現在，筆者のまわりにはモデムが山積みされています。この春，出荷ラッシュであった各社FAXモデムの動作テストのためです。

最近，Macintosh用のFAX送受信プログラム，またパソコン通信ソフトの開発をしたためにこのような状況になってしまいました。本編とは関係ないのですが，なんとかX68000上でもFAXモデムを動かしたいと思う今日この頃です。

さて，次回はモデムづいていることもあり，趣向を変えてModem⟷Modem経由の電話による遠隔データ転送を考えてみようと思います。

図1

(a) データ長8ビットのときの送信タイミング

実際のデータが送られるタイミング。長さ8ビット

信号レベル HI

bit0 bit1 bit2 bit3 bit4 bit5 bit6 bit7

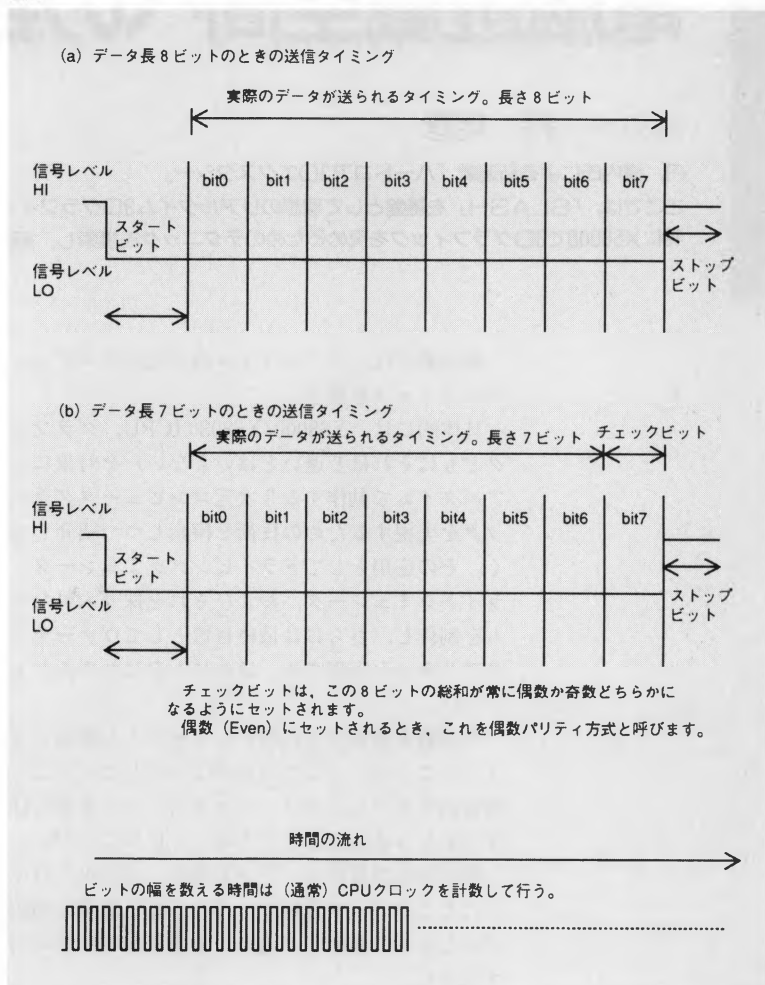
スタートビット

ストップビット

信号レベル LO

チェックビットは，この8ビットの総和が常に偶数か奇数どちらかになるようにセットされます。
偶数（Even）にセットされるとき，これを偶数パリティ方式と呼びます。

時間の流れ

ビットの幅を数える時間は（通常）CPUクロックを計数して行う。

Macintoshに関するいくつかのこと

最近，X68000とMacintoshを併用する場合が多いようです。しかし，同じMC68xxx系のパソコンでありながら，これほどファイル変換の厄介な組み合わせはありません。

それぞれの対応しているフロッピーディスクを見ると一目瞭然です。

最近のMacintoshは2DD（720KB），2HD（1.4MB）をハードウェアの段階で読み書きできます。これはIBM PCとのデータ互換を確保するためにアップルが歩み寄ったということだと思います。

また，OSのファイル管理方式が独特でDOS系（X68含む）と一切互換性がありません。テキストファイルと画像のTIFF形式以外は互換を維持できないと思ってください。

結局のところ，Macintoshとのデータ互換は市販の通信ソフトによる転送でしかできないと思ってください。サンデーエンジニアリングで行うにはハードルが高いようです。

X68000	Mac/SE30以前のモデル	MacIIci以降（現在の標準的モデル）
5"2HD（1.2MB） 3.52HD（1.2MB）	2DD（GCR形式800KB）	2DD（GCR形式800KB,720KB），2HD（1.4MB）

注，GCR形式はMacintoshだけが採用している方式。
ディスクの回転の制御方式が通常と異なります。ハードウェアの段階で互換性がありません。

SIDE A

電脳遊戯空間への野望

Tan Akihiko 丹 明彦

丹，横内氏による新連載「ハードコア3Dエクスタシー」
ここでは，「SLASH」を基盤として理想のリアルタイム3Dグラフィック環境を目指す
特にX68000で3Dグラフィックを究めるためのテクニックを模索し，紹介していく

本連載では，リアルタイムの3次元コンピュータグラフィックを扱う。

具体的には，X68000/X68030（CPU，グラフィックともにそれほど速いとはいえない）を対象に，リアルタイムで動作する3次元コンピュータグラフィックを実現するための技術を模索しつつ紹介していく。その応用としてドライビングシミュレータ，フライトシミュレータ，およびそれを採り入れたゲームを制作し，さらには最終目標としてヴァーチャルリアリティを実現する。ほかにもなにかやるかもしれない。

本連載を開始するにあたってぜひとも強調しておきたいことは，ここで実現しようとしていることは最新のグラフィックワークステーションを使えばできてあたりまえのことである，ということだ。

我々の武器はアセンブラである。爪に火を灯すようにしてクロックを稼ぎ，はしょれる計算は極限まではしょり，容赦なく迫りくる垂直帰線期間に対抗するのだ。

しかし，ここまでしても，テクスチャマッピング時で何百万ポリゴン/秒，などという性能を競い合うグラフィックワークステーションの世界で，Cかなんかを使ってへろへろと書いたアプリケーションの足もとにも及ばないことを思うと，ときどき生きているのが嫌になるが。

X68000/030は，確かに素性のいいアーキテクチャをもったマシンではあるのだが，どちらかといえばスプライトばしばしのアーケードゲームの移植でのみ，その実力を認められてきた感がある。特に重たい3Dフライトシミュレーションなどといった場面では，CPUの非力さを露呈してきた。

それでもX68000/030は真にパーソナルな意味でコンピュータを使わせてくれる最後の機種であるということを信じて，前向きにやろうじゃないか，ということ。

そして啓蒙という意味では，高級言語（またはそれに準ずるもの）によってアルゴリズムを見通しよく解説することも必要と考える。そのための言語として，Cを選ぶ。これはある程度速度を度外視したものでもかまわない。ひとりでも3Dフリークを増やすための餌である。

連載開始

連載は，パワフルなプログラマにしてゲームにも一家言をもつ横内威至氏との分担で進める。横内氏は今月号の付録ディスクで発表したポリゴナイザライブラリSLASHの作者である。当然，彼がSLASHシステム全般の解説を担当する。

実装に即した高速なアルゴリズムの構築と，アセンブラによる硬派プログラミングをたっぷり味わっていただきたい。それから，SLASHシステムの応用のための理論とCによる軟弱プログラミングを，私が担当する。ゲームやヴァーチャルリアリティの段階に進むと，どうしても音師が必要になるのだが，それはそのときに考えよう。

連載にあたっての心構えとして3つ挙げておく。

まず「毎回なにかを動かす」ということを宣言しておきたい。今回は付録ディスクに実物があるのだから当然だが，次回以降も理論に終始することは避けたいと考えている。

そして「ノウハウを伝承する」というのも大切だと考える。3次元コンピュータグラフィックの手法，それを実用的な速度で実現するための技，そしてアセンブラこそ美しいという思想。

そして，最後に「文体はクールに，内容はハードに，精神はホットに」を合言葉として，資料として残せる美しい連載にしたいと考えている。

今後の展望

今回はご挨拶であるから，理論的な部分には立ち

入らない。付録ディスクに収録したサンプルプログラムに触れ，SLASHのポテンシャルを感じとってほしい。今回はライブラリの使い方についても軽く解説するので，できる人はどんどん進んでいってもらいたい。

最初にも述べたが，当面の目標は3次元のリアルタイムシミュレーションである。これを実現するための要素について考えてみる。連載の性格上，コンピュータグラフィックの理論書レベルのことは押さえつつ，高速化を念頭に置いておく必要もあることを覚えていてほしい。

1） データ構造

我々はすでにSLASHシステムを手にしている。これは取り扱うデータ構造が決まっており，したがってここで議論する必要はあまりない。ただし，連載が進むにつれてより高速なアルゴリズムなどが発見される場合もあり，そのときはSLASHシステムも変貌し，加速することだろう。これはシステムの性格を考えれば当然のことである。いつまでも仕様が安定しないなどといってクレームをつけないようにいまからお願いしておく。

2） 座標変換

SLASHシステムの目標は，画面の真ん中にポリゴンの立体が出てきてくるくる回るのを見て，ああよかったね，で終わらすことではない。やるべきことは，あくまで3次元リアルタイムシミュレーションである。

たとえばドライビングシミュレーションなら道路を車が走る。フライトシミュレーションなら空を飛行機が飛ぶ。宇宙戦争ものなら（今回のSION IVを見れば明らかだが），宇宙空間があり，隕石があり，戦闘機があり，巨大戦艦がある。

これらはある一定の規則のもとで空間内を運動する。この状況をモデル化するためには，それぞれの物体に座標系を設けるというのが一般的な戦略である。そして，それぞれの物体を適切に座標変換することで画面に表示する。

実際のところ，SLASHシステムは表示だけでなく，座標系や座標変換の部分も面倒をみる仕様となっている。ただ，現時点では，後述する力学シミュレーションで起こりうる複雑な座標系と座標変換にまで対応しきれるかどうかが不明であり，ここで理論武装することは無駄ではないと考えている。

3） 描画

この計画を開始した当初は，私も表示にまつわる理論を展開し，また実践もするつもりだったのだが，横内氏の仕事があまりに速いので，全面的に任せることにして，私はその成果をありがたく利用することにした。X68000のアーキテクチャをしっかりとつかみ，アセンブラの癖もよくわかっている氏は，現段階までに驚嘆すべきシステムを作り上げた。私のやったことは，これをC言語からコントロールできるようにささやかなプログラムを書くことだけである。

4） 力学シミュレーション，シミュレータの実装，ゲームデザイン

いま私の興味は力学シミュレーション付近に向いている。海外には，力学シミュレーションを相当きちんとやっているのに，いやそれゆえに高いゲーム性も備えているドライビングシミュレータが存在する。

何度となくほめたたえてきた「WORLD CIRCUIT」や「Indianapolis 500」などは，そんじょそこらの疑似3Dアーケードゲームを残らずカスだといいきってしまえるほどの素晴らしさだ。両者とも，AMIGAシリーズだけでなく，IBM PC/AT向けにも発売されているので，すぐにでも味わってみたいという方は入手することをお勧めする。はっきりいってこのレベルに達した国産ソフトはまだない。

ひょっとして日本のパソコンユーザーたちは，本当に車を運転しているみたいな感覚に陥らせてくれる，ドライビングシミュレータに触れたことがないのではないか。もしそうなら，ここでぜひともホンモノを作り上げ，ドライブゲームにおけるスタンダードたるべきものを提示したい。これが本連載の動機といってもよい。正直いって，あのレベルを実現するのは非常に難しい。しかしそれゆえにチャレン

ノウハウの伝承

近年の世界的な流れとして，知的所有権というものが過剰といっていいほど認められている。連中は，プログラムは知的活動の産物だからといって著作権を主張し，その同じものに対して工業製品であるという理由から特許権も主張する。著作権を主張する割にはソースコードは公開しない。特許権を主張する割にはノウハウを隠蔽し，リバースエンジニアリングも禁止している。そしてその知的所有権を侵害した者を告訴して利益にしようと躍起になっている。というか製品を作るよりそのほうが儲かると気づいて得意になっているかに見える。

昔からのソフト屋のひとりとしてはいやな時代になったと感じるし，その直感は間違っていないと思う。

いや，私はなにも市販ソフトを解析してソースコードを盗めとか，プロテクト破りをしろとかいっているわけではない。ただ，モノ作りの基本も知らないくせにコンピュータが金になると気づいた連中が，あまりに破廉恥なやり方で権利を主張しているという現実に嘆きを禁じえないのである。こんな時代では，たとえばGNUプロジェクトのようなフリーなソースコードを公開する貴重なプロジェクトは早晩崩壊してしまうだろう。

将来，痴的もとい知的所有権がらみの裁判に対策できる余裕のある大企業の庇護のもとでしかプログラムを作れなくなるとしたら，それはとても悲しいことだ。

ジングなテーマなのである。

とりあえずの制作予定として,「首都高速ドライビングシミュレーション」,横内氏が入れこんでいる「ル・マン24時間」が挙がっている。私自身はモーターサイクルのシミュレーションもターゲットに入れている。

私はF-1のファンであるが,F-1は制作予定に入れていない。フジテレビがF-1のゲーム化権を牛耳っているからだ,というのは半分冗談だが半分本気である。これは,私が個人的にフジテレビ3つの大罪と呼んでいるもののひとつである(ちなみに残りの2つは,F-1をバラエティ番組におとしめたことと,TV版の総集編をビデオなりLDなりで発売していないことである)。

極東の1テレビ局が,全世界のソフトハウスの行動を縛ろうなんて行為はいただけない。事実,昨年の後半以降発売されたF-1ものには,AMIGA, IBM PC/ATを問わず,フジテレビの名前がしっかりと入っている。前にもいったが,ゲーム化権をテレビの放映権と一緒に考えてもらっては困るのだ。

そして,これを「フジテレビオフィシャル」とかいってありがたがる輩がいるというのも困りものだ。私はこの件以来,「オフィシャル」という言葉が大嫌いになった。

5) その先

リアルタイムシミュレータを実現したその先は,ご存じヴァーチャルリアリティ(VR)へ進みたい。VRという単語自体がもうあまりにメジャーになりすぎて陳腐化しつつある昨今,この単語を使うのもなんとなく気がひける。が,そのコンセプト自体は依然として新鮮であり,その恐るべき可能性については明らかになってはいない。

VRの工業応用とか商業応用には基本的に興味はない。VRは「仮想現実感」などという訳がつき,データグローブでアクセスする虚構の空間というイメージができているのではないかと思うが,Virtualという単語が「事実上の」という意味を本来的にもっていることを考えると,むしろ「現実よりリアルな現実」の線を目指したくなる。横内語録から拝借するなら,まさにドラッグになりうるものを作りたいのである。

SLASHシステムの行方

SLASHシステムは,この連載の核といえる。その狙いのひとつは3Dフリークの発掘にある。

SLASHシステムを利用する際にはある程度の資格を要求したいとも思っている。それはつまり,3Dシステムに対する信条とか美学とかいうものだ。それは連載の中から汲み取れることと思う。自分がこちらの世界の人間であると思った方は,がんがんついてきていただきたい。

いま,我々は,X68000上で3Dシステムを構築するうえでの最大のネックであった「実用的な速度で動作するポリゴナイザ」を手にした。もう,やるしかないではないか。

[付録 _slashlibガイド]

0.一般的なこと

以下に記述することは,SLASH ver.1.0(今月号の付録ディスク収録バージョン)のCフロントエンド_slashlibに関するドキュメントである。本来SLASHシステムは,アセンブラレベルで利用したときに最高のパフォーマンスを発揮するパッケージであるが,_slashlibパッケージはSLASHシステムをC言語から扱いたいという軟弱なプログラマ(私も含めて)のためのものである。

_slashlibパッケージは正しく動作することを期待されて制作されたが,その使用頻度に応じてテストされていない関数もまだ多数存在する。また,SLASHシステムが完全に正しく動作するということを前提にして制作されている。現段階では,SLASHシステムに由来する不具合はないはずである。

1.ファイル

“_”(アンダースコア)で始まる名称は,変数名や関数名,ファイル名ともにC言語に関連することを示す。

slashlib.a	SLASHシステムの座標変換部分と通常形式描画部
slashRlib.a	ラスター抜き描画部
slashBlib.a	デバッグモードつき描画部
_slashlib.a	Cのフロントエンド関数
_slashlib.h	Cのインクルードファイル
_tpllib.a	標準色ライブラリ
_tpllib.h	標準色のインクルードファイル

2.コンパイル

基本的には次のようにする。Cコンパイラが正しく動作するための環境は作っておくこと。

```
gcc foo.c -lcolor -l_slash -lslash
```
通常形式の描画
```
gcc foo.c -lcolor -l_slash -lslashR -lslash
```
ラスター抜きの描画
(-l_slashは-lslashよりも先に書く)

ラスター抜きバージョンは理論的には通常バージョンと同居できるはずだが,ライブラリのサイズがMC68000のPC相対の限界を超えてしまうために,

今回はとりあえずラスター有/無バージョンは排他的にしか存在しない。関数名やラベル名はslashlib.aとslashRlib.aとで共通。

3．データ型

_slashlibで用いるデータ型は_slashlib.hで構造体として宣言されている。詳しい内容は_slashlib.docまたは_slashlib.hを参照されたい。むろん横内氏によるSLASHそのものの説明がもっとも有用であることはいうまでもない。

4．_slashlibの使い方

以下，最低限の注意事項を示す。サンプルのobjtest.cがすべてを語ってくれるはずである。

4.1 物体を作成する

これには2とおりのやり方がある。
1) SLPOINTLIST, SLPOLYGONLIST型の変数を宣言してユーティリティ関数などを用いて物体を作り上げる

SLPOINTLIST, SLPOLYGONLISTのいずれも構造体としてのサイズはそのヘッダ部分しかなく，残りのデータ本体部分はsizeof (SLPOINTLIST)で得られるサイズの中には入らない。したがって，変数の宣言はポインタによって行い，必要な領域はmalloc()関数によって確保する必要がある。確保するサイズは，それぞれ次のようになる。

ポイントリスト　sizeof(SlPOINTLIST)＋(頂点数)×sizeof(SLPOINT)

ポリゴンリスト　sizeof(SlPOLYGONLIST)＋(ポリゴン数)×sizeof(SLPOLYGON)

ユーティリティ関数は，addpoint()，addline()，addtriangle()，addtetragon()およびmakebox()。これらはポイントリストとポリゴンリストを自動的に更新する。同じ頂点を2個以上作らないようにチェックする機能もある。
2) オブジェクトファイルの形で別途アセンブルしておき，externで参照する

これは主にモデラで作成した物体を利用する場合に用いる。領域確保などの手間がかからない。付録ディスクのフォント集のように，ライブラリ化することも可能。

ポリゴンマクロは宣言して2つまたは3つのポリゴンリストを登録すればよい。また，ポリゴンマクロは必須ではないので，ポリゴンリストが表示できないうちは使わないほうが無難。

ポリゴンの色は，標準的なものをライブラリとして用意しておく。1) の場合，色は自由に決められる。2)の場合はモデラがデフォルトの色を記録している。いずれも，標準色ライブラリをリンクすればよい。

各物体に対して，AddNorm()関数を施して，法線を付加しておくこと。シェーディングのための大切な情報になる。必要ならば，SortPoly()関数でポリゴンをソートしておく。

4.2 各種変数の宣言をする

SLTRANSWORK型の配列を宣言する。配列の要素数は頂点数に等しい。

SLMINIMAX型の変数の宣言は特に説明が必要である。まず2組必要というのがポイントだ。グラフィック2ページを切り替えると，ちらつきを防ぐ効果がある。片方のページを表示しておいてもう片方のページに描画する。描画が終わればそのページを表示して，もう一方の(それまで表示されていた)ページに描画する。SLMINIMAXワークエリアは描画した物体の消去に用いるためのバウンディングボックスの情報である。それは描画時に構築され，消去時に使用される。したがって，約2フレームの間は内容が保存される必要がある。このためにSLMINIMAXワークエリアは2組必要なのだ。このほか，SLPALAMETER型も宣言しておくこと。

4.3 物体を表示する

まず前準備として，描画ウィンドウをSetWindowSize()関数で設定する。次に描画ウィンドウの中心をSetWindowCenter()関数で設定する。そして消去色をSetClearColor()関数で設定する。

以下は毎フレーム行う処理である。SLPALAMETER型の変数の値は設定すること。上でも述べたように，ページ切り替えを駆使する。

まずSetWritePlane()関数で表示開始位置を決める。この値を毎回交互に各ページの左上の座標に切り替える。

複数の物体を表示するとき，テンポラリのSLMINIMAX型へのポインタをひとつ用意しておき，現在のフレームで用いるSLMINMAXワークエリアへのポインタを代入しておく。これは重要である。

以下は各物体に対して行う。

TranslateAll()関数で，各項点を座標変換し，結果をSLTRANSWORKワークエリアに格納する。

DisplayPolygonList()またはDisplayPolygonMacro()関数で描画を行う。

AdjustMinMax()関数でSLMINMAXワークエリアの補正を行う。この関数の戻り値を，テンポラリのSLMINIMAXワークエリアへのポインタに代入し，次回はこれをもとにTranslate()，DisplayPolygonList()またはDisplayPolygonMacro()およびAdjustMinMax()の処理を行うこと。この手順を守らないと消去が正しく行われない場合がある。

SIDE B

ヴァーチャルドラッグを超えろ

Yokouchi Takeshi 横内 威至

常に熱い刺激を切望する横内氏が提唱するバーチャルドラッグ
それは，仮想現実の枠を超え，現実以上のリアリティともいえる
この野望を実現するためにどうすればいいか，なにが必要かいま一度考えてみよう

絶頂のヴァーチャルドラッグ

若僧は刺激に飢えている。俺だってそうだ。車に乗るなら限界までエンジンを回転させ，高まる鼓動にテンションを高める。路面に食いつけなくなる瞬間までステアを当て，弾き飛ばされるほどの強烈な重力を体で食い止める。こんな強烈な刺激は，思考，理性さえ捨て去れば誰にでも魅力あるものだ。だが理性あるかぎり抑え込まねばならない欲望なのである。事故を起こせばすべてを失い，人を挨ねれば一家を心中に陥れる可能性を十分含んでいるのだ。

また，刺激を求めて人は映画に魅了される（俺はいまさら面白く感じない)。映画の刺激は確かに素晴らしい。圧倒される派手なアクション，押し寄せる現実的でリアルな映像，誰でもハイになれるものだ。もちろんロマンスだとかは話にしてない。ところで貴方は映画を見ている自分の立場を考えたことはあるか？　貴方の存在はまったくない。かぎりなく主人公に近い視点とはいえ，あきらかに違う。「俺だったらこんな愚かな行動はとらない」，誰もが一度は考えたことがあるはずだ。映画はどんなに進化しようとも，もう映画の壁を越えることはできない。

そこでゲームだ。まだ一般に受け入れられているとはいい難いが，あらゆる可能性を秘めている。シミュレーションと呼ばれる，現実を可能なかぎり再現した仮想空間は，ある程度脳を騙すことでいくらでもリアリティを感じさせる。人が現実世界で求める刺激に近い感覚をコンピュータは作り出し，人の脳を掻き乱す。これはヴァーチャルリアリティの世界である。

俺はリアルな刺激を求めている。リアルでないものはどうしても嘘臭さが鼻につき，ハイテンションから一気にひきずり落とされる。その点，映画は結構いい線いっている。だが映画自体には限界がある。一方的なメディアでしかないのだ。だがゲームは無限の可能性を秘めている。映像こそまだ映画には負けるが，いつかは必ず映画を超越する。ソフトウェアの世界はまだ限界を明らかにしていない。

そうだ。遊んでいるヒマなどない。すでにいまある現実に飽きてしまった我々は，この真新しいドラッグを自ら作り上げねばならないのだ。

ジャップと3D環境

さて，ヴァーチャルリアリティとはどういったものであろう。ヴァーチャルリアリティとはいたって高い壁である。基本は我々の存在する3次元空間である。まずこれをコンピュータで制御するのが基本である。だが，あらゆるリアリティを追求するには，狂ったほどリアルなインタフェイスが必要だ。当然このために，いわゆるオブジェクト指向なんてのも採り入れることになろう。つきつめれば本来原子レベルまでをシミュレーションしなくてはならない，コンピュータとして究極の課題のはずである。だが我々はそこまでやらない。プロの研究者ではないのだ。我々は，その基本となる3Dのとっても美味しい部分を追求するつもりである。要するに3Dモノと分類されるジャンル，シミュレータとでも思えばよい。

現状を見てみるとジャパンの3Dモノは，まだまだ二流だ。最近流行りだしてはきたが，しょせん海外のトップクラスと比べるとハナクソにもならない(アーケードクラスとなればむしろ逆かもしれないが)。グラフィックワークステーション並みのポリゴナイザを装備していれば，巻き返しも可能なのかもしれない。海外のトップレベルを知っているか？　ハードディスクベタ読みの垂れ流し，CD-ROMベタ読みの垂れ流しなんていう，志のない，3Dの意味のないことはまずやらない。見かけ倒しはやらず常に本物のリアリティを追求しているのだ。そして海外の真の凄さは，すでに処理の速さ自体で勝負していないところにある。それを制御する領域で勝負して

いるのだ。当然タイヤは路面を噛み，本当にリアルなグリップ感を表現しているのだ。バーチャレーシングは確かに凄かったが，車の挙動を感じるにはいたっていない。やはり海外とのレベルの差は大きいらしい。

我々は可能な限り3Dを追求したい。現実にはその教科書もなく，自力で考えていかねばならないため道は険しい。そして本来CPUパワーで押せば多少の重い処理など問題にならないのだが，X68000となるとこれさえも大きな課題となる。勘違いされると困るのだが，ポリゴナイザがあるからといって3Dモノを簡単に作成することは不可能なのである。MAGICがあったからといって貴方にSIONが作れるか。3Dシステムとはいっても，しょせんスプライトルーチンを作った程度でしかない。さらにそのうえでゲームという理論を理解しなくては不可能なのである。

モノによっては，より相性のよい3Dシステムを必要とすることも忘れないでほしい。貴方のやりたいことによってはポリゴナイザの設計からスタートしなくてはならないのである。このあたりも含めて，我々はいっさいを捨てて突っ走らねばならない。

ソフトウェア技術の必要性

俺はまず道具を作った。SLASHシステムである。なんのノウハウもなかったが，考えられるかぎりの高速化を追求し，さらに独自のアルゴリズムでシェーディングを可能にした。いろいろと世間にはポリゴナイザが出回っているようだが，それらより一段上をいくよう極力効率を上げているつもりだ。

だが，本来こんな努力はCPUさえ速ければ，あるいはまともな専用チップ，たとえばAMIGAのブリッターハードのようなものでもあれば，必要性は失せるのである。X68030のパワーはそこそこある。だがそのハードは68030のパワーを有効に活用できない。グラフィックは10MHzで動いているのだ。そして大容量のグラフィック処理，たとえばクリアという作業は大量のデータを書き込まねばならないのだが，基本的にはなにも考えられていない。ハードウェアクリアもあるが，まともに使えるものではないのだ。どうする？　我々はハードに期待できない。ソフトですべてを解決しなくてはならないのだ。これを予想してハードを作れなかったのだろうか？まあしかたないか。だがそのくせCPUはもう遅すぎる。そしてハードはCPUの足を引っ張る。海外に遅れているのはソフトだけではない，ハードさえもあちらには劣るのだ。そうなのだ。ソフトを知らない者に優秀なハード，特に特別な思想をもったハードは絶対作れないのだろうか。AMIGAの思想は素晴らしかった。開発者がフライトシミュレータを熱望して作っただけあり，ポリゴンには非常に強力なパワーを見せつける。そしてアップル時代から培われた狂気の技術が乗っているのだ。先を見る目も確かである。CPUさえ換えればいくらでもスピードが上がる。俺がAMIGAによって学んだことは大きい。俺はもう日本のコンピュータには期待を抱かない。

まあとにかくいま我々が最も扱いやすいのはこのX68000である。弱音を吐いているヒマも許されない。いずれくる新たな世代でもソフトウェア技術は要求される。特にアセンブラプログラマであれば，いくらシステムがハードウェアになってもその動作だけは絶対把握する。結局，どうやって動かしているかはソフトウェアの技術である。技術は積み上げていくものであり，もしここでやらなければコンピュータはもう我々に手の届かないレベルにいってしまうであろう。どんなにハードが進化しようとも，ソフトの技術は絶対必要。いかにDOS/Vマシンが速くても，最先端のソフトウェアは，常に異常な技術で支えられていることも忘れてはならない。せめて我々はその基本だけでも押さえなければならない。半分危機感もある。いまついていけなかったら，我々はもう素人の遊びのレベルでしかないのだ，と。

展望と野望

研究する基盤はできた。ごく基本のポリゴナイザである。これをベースに我々はいろんなタイプの3Dモノを扱っていきたい。いちばん簡単なのはスターフォックスタイプ。座標系なんてあまり重要でないし，やれば力で押せる。あ，でもゲームって奴をしっかり作れない人には絶対不可能。このタイプは俺よりもSIONシリーズプロデューサの山田氏のほうが，よっぽど先を走っている。まあ，チャンスがあれば俺も一度は手を出してみたい。

そしてフライトシミュレータである。似たようで相対するモノとしてはドライビングシミュレータ。どこが違うのかはいずれ明らかになっていく。これらは2つの大きな問題点がある。ひとつは座標系。回転というものは恐ろしく複雑なもので，見方によっていくらでも変わってくるのである。いったいどういった表し方ができるのか，どういったモノが有効かはいずれ述べてゆく。もうひとつは制御システム。いかに本物に近づけるかはもう完全別問題。

実はポリゴナイザの制作のきっかけは，ドライビングシミュレータを制作しませんか，と中野氏からの甘い誘惑であった。ラスターモノでどれだけ可能かを最初は考えていた。ラスターモノ，といえばアレだからだいたい時期は想像できるであろう。そのうち本当にリアルなモノを考えると，どうしてもしっかりとした演算が必要だと方向がずれていく。か

つて一度だけ組まれた3D特集，あの頃から崇拝していたプログラマ丹明彦氏に話をもちかけて結構真面目にスタートする。そしてポリゴナイザである。相当の時間がかかっているが，ようやく「SLASH ver. 1.0」として一段落できるものとなった。

おっと，話がそれたがつまり俺は，イカしたハイスピードドライビングシミュレータを制作していくつもりなのである。これはサンプルのテスタロッサを見てもわかることであろう。ちなみに俺はマイナーフェチだからF1よりもル・マンを愛する。だけど東京にいないと本来データが取れないような奴を予定している。以上が野望であるが，現段階で30%ぐらいは見えているといっておこう。

SLASH

さて，SLASHであるが，いまいちユーザーには冷たいシステムである。最初のステップをしっかり踏まえなくてはまったく使いこなせない。MAGICを使っていた人であればある程度たやすく移行できるが，モデリングで結構つまずく。面ソートは基本的には行わないため，データ作成でうまく順序だてなければならない。

また，なにより困るのは画面クリアだ。ということで今回はある程度のリファレンスを行う。

基本となる流れを図1に書いておく。一般的なループであればそのとおりの手順になるはずである。もしかするともっと有効なテクニックがあるはずだが，まだはっきりとは解説できない。

●ウィンドウについて

まずウィンドウ設定であるが，デフォルトは256×256である。現段階ではこれ以上のサイズはサポートしていない。このシステムはリアルタイム制御を目的としているため，解像度を優先的に犠牲にする。もし希望さえあればさらに上のものを作るかもしれないが，同程度の画像のために，単純計算で4倍の遅さになってしまう。これはむしろラスター抜きポリゴン描画をメインにするぐらいの覚悟はしておくべきである。まして10MHzではあきらめるしかないレベルかもしれない。

ウィンドウサイズとは別に消失点を定められる。基本的にはウィンドウ中心座標を設定してもらいたい。だが低空飛行なんかをやると，視界不良がかなり目立つ。そんなときにはやや上方にずらすとよいかもしれない。なんとなく視点の移動っぽく感じられるが，視点移動のためにはまず使えない。試しにウィンドウ外に大きく外してみると，かなりマズイ画像になってしまう。視点移動に関してはまた別問題として扱わねばならない。いずれ座標系について説明するときにこの辺のことも含めることにする。

そしてクリアカラーであるが，ほとんどの場合は0を設定すべし。0以外の値にしてしまうと，グラフィックより優先順位が低いモノは表示されなくなってしまう。なお，デフォルトでは灰色である。

●描画エリアについて

描画アドレスは任意のアドレスを指定する。このアドレスを左上とするよう，ウィンドウがセットされる。下手なアドレスにすると，当然バスエラーになってしまう。また，本来望ましくないが，もしX68030で10000枚ものポリゴンを表示させる必要が出たときは，メインメモリをバッファとして取り，最後にグラフィックに転送するほうが速いかもしれない。グラフィックが10MHzと遅めであるからだ。正しくSX-WINDOWで動かすときにも，メモリ上にバッファを取るべきであろう。

クリアに関してだが，場合によっては全画面クリアのほうが速くなる。ポリゴンが画面全体に及んだりすれば一気に動作が遅れてしまうためだ。もし異常に多くの物体を表示するのであれば，割り切って自前のクリアルーチンを用意すべきだ。SLASHで

図1

```
初期設定：①ウィンドウサイズ認定
          ②ウィンドウ中心設定
          ③各ポリゴン法線設定
          ④ミニマックスワーク0，1にエンドコードを書く

全体のループ：
  ┌ バンク0を描画アドレスとして設定                 →SETWPLN
  │          ┌M，Mワーク0を指定して座標変換       →TRANSLATER
  │ 物体ループ┤M，Mワーク0を指定して表示           →DRAWPOLY (SORT)
  │          └M，Mワーク0を指定してアジャスト     →ADJUSTMINIMAX
  │ バンク0を表示                                   →任意の方法
  │ バンク1を消去アドレスとして設定                 →SETCPLN
  └ M，Mワーク1を指定してクリア                     →CLEARBOX

  ┌ バンク1を描画アドレスとして設定
  │          ┌M，Mワーク1を指定して座標変換
  │ 物体ループ┤M，Mワーク1を指定して表示
  │          └M，Mワーク1を指定してアジャスト⇨同上
  │ バンク1を表示
  │ バンク0を消去アドレスとして設定
  └ M，Mワーク0を指定してクリア

  ループ先頭へ

物体ループ：
          movea.l  CWORK, a3
          jsr      TRANSLATER
                   ：
          movea.l  CWORK, a4
          jsr      DRAWPOLY (SORT)
                   ：
          movea.l  CWORK, a0
          jsr      ADJUSTMINIMAX
          move.l   a0, CWORK
            ループ先頭へ
```

は物体ごとの部分クリアを前提としているため，このまま全体クリアに切り替えると結構な無駄を出してしまう。次のバージョンでは，部分クリアに関する部分をいっさい省くモードも用意すべきであろう。

●座標変換について

さて，座標変換であるが，現在では単純にX，Y，Z軸回りの角度で指定するしか方法がない。それぞれPITCH，HEAD，BANKということであるが，正確にはまったく別物の定義であろう。正確にこれらを表すにはオイラー角で表さなければならない（オイラー角はいずれ解説していく)。もし本当に有効な座標系であればシステムに積むことになるであろう。

また，物体のオフセット座標に関しては自前で用意しなくてはならない。これはちょっとした変更ですぐ可能なのだがまだついていない。申しわけない。これができないと本当は地上のポリゴン表示なんかは苦しいのだ。いずれ必ず加えることにしよう。

変換バッファは1ポイントにつき8ワード。ひとつの物体で最大2048個である。理由は明快。16バイト×2048＝32768，つまりワードサイズのインデックスつきアドレスレジスタ間接によってポイントできる最大値である。実用性から見れば十分であろう。図1では同じワークをすべての物体で使っているが，もしワークを保存しておくなら任意のワークを与えればよい。

●クリアについて

特別企画の記事のほうで具体的な動作を説明しているのであまり触れない。しっかりと把握しておいてほしい。とりあえずテスタロッサのサンプルでは複数物体には対応していないので，図1の一般的な物体表示，クリアルーチンを解説する。図1のようにいきなりループの体制に入るなら，クリアワークにはあらかじめエンドコードを書いておかなければならない。流れを見ればわかるであろう。

●表示について(1)

基本はポリゴンマクロよりポリゴンリストである。単純に与えられたデータどおりに上書きしていくだけである。ソートなしで不都合がないかを考察してみよう（図2)。

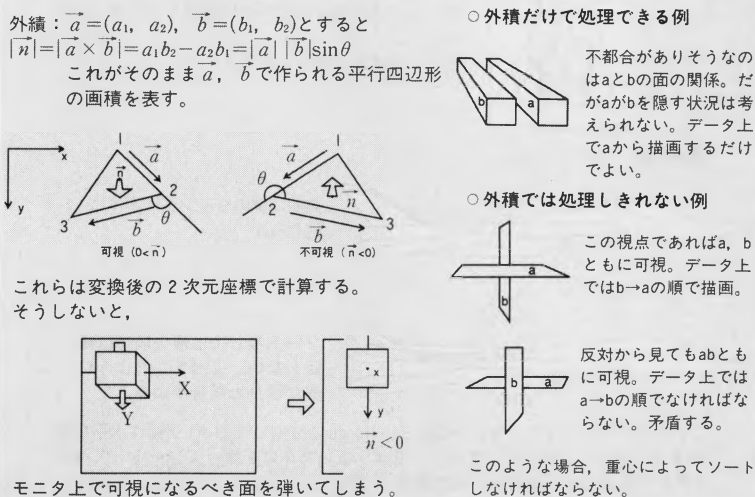
まず単純な立方体を考える。裏になった面をいっさい表示しなければ，どんな順番に面を描いても問題ないことは明らかだ。裏返った面の判定はベクトルの外積を使う。外積の定義は中学レベルのはずである。要するに2つのベクトルに垂直で，大きさは2つのベクトルで作られる平行四辺形の面積に等しい。まずは，2次元で考えよう。もし2つのベクトルが平行ならば面積は0。大きさはベクトルのなす角度によって符号が変わる。この座標系であれば正なら手前を向いており，負なら逆。つまり負ならば表示をしないということである。危険性を考慮し，0のときの表示を行っていない。

ここで陥るトラップは2次元変換後の座標で判定しなければならないことである。3次元のままで弾くと無理がある例を図2で示しておく。そして，2次元座標で問題となるのはポリゴンが小さく表示されるときである。縮小表示されると，本来ベクトルを表す2点が重なってしまうことが考えられる。こうなると片方のベクトルは0であり，実際には正面を向いている面さえも弾かれてしまう。ならば0を弾くのをやめればすむか，といえば甘い。今度は裏の面が表示される危険性が出てくる。どちらかというと裏が見えてしまうほうが画像として美しくない。よって現在の方法にしてある。

さて，今度はどんなときにこのソートなし描画が困るかをやはり図2で考えてほしい。まず貫通体は明らかに駄目。これはモデリングが悪い，ということにして物体を分割する。それでも問題は出るのであるが。

そこでマクロソートである。まず単純な面ソートについて考えたいが，これはいずれZバッファアルゴリズムを解説する予定でいるので今回は流す。普通に考えられるソートであれば，面の重心を計算し，視点からの距離で(Z座標の大小では不都合がある）判定する。これはあまりに重い処理なのでまずソートを避けているのである。ただし，物体がトランスフォームなどをしないかぎり，それぞれの面の重心は決まった座標で表すことができる。これはあらかじめポイントとして計算しておくことで，重心を求める計算を省くことが可能であろう。

では物体どうしでのソートはどうだろう。やはり同じである。そして重心といっていたが，それに似たような関係をもつほかの2点でもある程度は代用できる。図3である程度の解説をしておこう。サン

図2

プルのテスタロッサを考えてほしい。本来ひとまとめて描画するにはかなり無理がある物体であることは想像できる。

まあ，やりようによってはそこそこいけるのだが(最初は本当にそうしていた)，やはり不都合が大きくなってきた。面はもう人間の頭では処理しきれないほど複雑な関係をなしている。そこでこれは上部とボディに分けてある。ちょっと考えればボディだけならソートなしでほとんど可能。また上部もリアにある窪みだけ考えれば普通の凸物体同様である。テスタロッサのサンプルをいじり，上部のみ，本体のみの表示を行って確認してほしい。慣れてくればどんなモデルが不都合を生じるか，だいたいつかめるはずである。

●表示について(2)

さて，シェーディングであるが，やりやすいのは65536色モード。カラーコードだけで考えればよいのだ。ただし本当にしっかりしたものを作るのならば256色モードを使用するべきである。そうしないとグラフィックプレーンは2枚使えないからである。256色モードでも色の選び方さえ心がければそう問題はないと思う。実際ゲームなんかでは256色モードでも十分芸術的な画像を可能にしているのだから。

たとえばハイライトの白なんかは，まったく同じパレットでもよいのだからうまくできるはずだ。まだカラーエディタなどがなく，実際には結構ツライかもしれないが。

シェーディングのためにはあらかじめ面に法線情報を与えなければならない。これはマニュアル計算ではあまりに非人間的であるため，一応サービスコールを用意してある。やや精度に不満があるが，動画になれば気になる誤差ではない。いまは物体すべてを最初に初期化しておくことが必要である。

この法線情報をいじるテクニックをちょっと解説しよう。サンプルのテスタロッサのホイールを見てほしい。これらは同一平面上であるが，明らかに光の反射はずれている。単純にこれは法線だけが違うのである。いまはまだこれを狙って作るのは厳しいと思うが，手はある。このデータ，最初はホイールの中心はややへこんでいる。このままでも問題なさそうだが，実は真後ろからだとかで見るとかなりヤバイ。そこでまず法線を計算し，そのあとで座標を書き換えている。たったこれだけである。

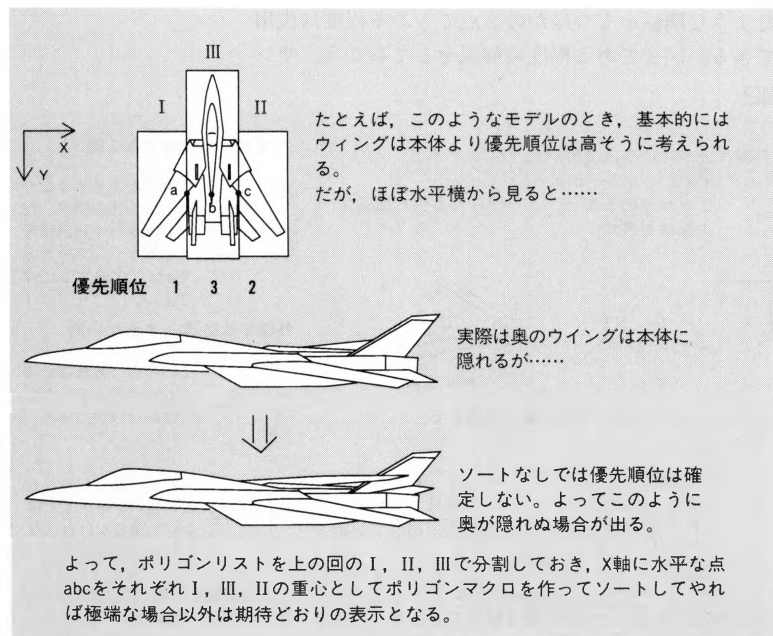
さあ，これでベタ塗りポリゴンから大きな飛躍をした。いまではほかのシェーディングのないポリゴンを見るとかなり不自然に感じてしまう。もうあとには戻れない。加速すること以外は許されない。メモリさえ許せばスムースシェーディングもできそうだがやっぱりかなりつらい。マッピングが欲しいが，普通に考えてもシェーディングと同居は厳しい。しかもCPUパワーを相当必要とするのであまり考えないでほしい。まずは，誰もやらなかったこのシェーデイングの効果を各自で味わってもらおう。

図3

予告

次回は，ポリゴナイザ自体を切り刻んでいこうと思う。実際のアルゴリズムを知ったうえで，完全な動作を把握しなくてはならない。それとともに，このシステム以外を設計するうえでの基盤となるアルゴリズムも追求していきたい。これらはもういい加減一般常識レベルにしてしまいたいのだ。SLASH自体でも3種類のアルゴリズムが存在している。もしかしたらX68030では，別のアルゴリズムが有効であるかもしれない。

俺もいよいよ生活が忙しくなってくるが，極力この3Dを究め尽くそうと燃えている。俺が3Dに本当にソソられたのは，Oh!X1989年7月号にあった3D特集であった。それ以来3Dが表立たないのにはかなり失望していた。もう俺が自力で切り開くしかない。本当にやりたかったシミュレータへの道をなにがあっても突き進む。なんのノウハウもなかった俺には異次元の内容であった。しかし，やってみればそこまで異常なものではなかったのだ。誰かがすでにやっていることである。俺だってここで負けているわけにはいかないのだ。基盤はできた。次は一気に制御を究めるのだ。快感のヴァーチャルリアリティは見えてきている。こんな熱いものを他人にばかり譲ってはいられないのだ。もう逆流するアドレナリンを俺は抑えられない，やってやるぜ。

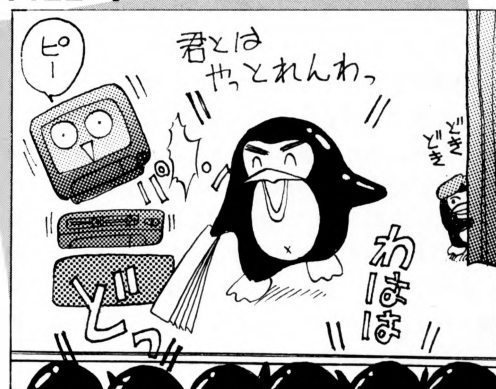
こちらシステムX探偵事務所

フッてボケてツッコんで

Itou Masahiko 伊藤 雅彦

今月は，なにやら新しい依頼人が探偵事務所を訪れたようです。どうやら依頼人は大阪からやってきた漫才師の卵。いったいどんな依頼となりますやら。そして，マスターと琴張夫妻に加え，探偵事務所初登場の伊藤氏の活躍はいかに!?

FILE-Ⅴ

illustration: T. Takahashi

依頼人（以下依）：うっうっ……。
マスター（以下M）：ま，まあそう気を落とさずに……。
琴張春香（以下春）：そうよ，人生には辛いこともあるけど，その先にはきっと明るい明日が待っているわ。朝陽はみんなを平等に照らし出してくれるの。
M：そ，そうそう。だから明るい明日に向かって……。
♪カラーン，コローン
琴張護（以下護）：ただいま戻りました。入口のベルは探偵事務所には不似合いですから，取ってしまったほうがよろしいと思いますよ。いつまでも昔のものに未練を残しているのは良くありません。
M：いまそれどころじゃないんです。
依：うっうう……。
護：おや，どなたですこの人は？
春：お客様よ。
護：依頼人の方でしたか。泣かせてほしいというご依頼だったのですか？
M：そんなわけないでしょう。常識で考えてくださいよ。
護：いや，この現代において常識にとらわれた思考ではものを見誤ることになります。歪んだ世相のなかで涙さえ忘れ，うつろな目をしている人間は少なくありません。泣きたい，涙を取り戻したいという依頼があっても決しておかしくありませんよ。
M：はいはい，そうですか。でもこちらの方はそうじゃないんです。
依：もうあかん，人生おしまいや……。
春：そんなこといっちゃだめ。あなたきっと売れっ子の漫才師になるわ。あたしの勘って，すっごく鋭いんだから。
護：ほう，この方のご職業は漫才師なのですか。大阪弁で漫才師とは，関西人のステレオタイプにピッタリとマッチしていますね。それでご依頼はなんでしょう。売れない漫才師を続けるべきか，やめるべきか，人生指南をしてほしいとおっしゃるなら，この私がお引き受けいたしましょう。決して悪いようにはいたしません。
M：余計なことしないでくださいよ。実はこの方，漫才の相方に逃げられちゃいましてね。捜して連れ戻してきてほしいっていう依頼だったんですよ。
護：なるほど，ありそうな話ですね。
M：それで，その相方の人を見つけるのは簡単だったんですがね，その人はもうほかの人と新しいコンビを組んでたんですよ。だから連れ戻すのは無理だっていまお話ししていたところなんです。
護：なんですと，その程度のことで依頼を反古にするのですか。探偵なら，連れ戻してこいといわれたら髪の毛を引っ張ってでも連れ戻してくるもんです。それぐらいの覚悟をもって仕事をするのでなければ，給料泥棒といわれてもしかたありません。
M：私はこの事務所の経営者なんだから，給料なんかもらってませんよ。
護：あ，そうでしたね。私としたことが，平静さを失っていたようです。ま，今回はお引き取り願いましょう。もしもしお客様，こんなところで泣いているより，早く新しい相方を見つけるか，漫談に乗り換えたほうがよほど生産的というものですよ。
依：でも，せっかくコンクールで優勝できると思てたのに……。
護：なんですか，コンクールとは？
春：えねーちけー新人漫才コンクールっていうのがあさってあるんですって。
M：そのコンクールでこの方のコンビが優勝候補といわれていたらしいんです。
依：これに優勝したらハクがつくし，テレビの仕事も入ってくるし，絶対に売れるはずなんや。
護：コンクールに優勝したぐらいで売れるとは思えませんがね。
依：……誰か相方やってくれまへんか。コンクールのときだけでええから。
M：そ，そういった依頼は……。
護：お引き受けしましょう。
M：ちょ，ちょっと琴張さん……。
護：お客様の依頼には極力お応えしなければなりません。
M：し，しかし，この依頼は探偵のすることではないんじゃないでしょうか。
護：この不景気なのに仕事を選んでどうするんですか。当探偵事務所はあらゆる依頼にマルチにお応えするのが信条なのです。
M：あなた，従業員のくせに勝手に経営方針を作らないでくださいよ。それに引き受けるったって，漫才なんかできるんですか，琴張さん。
護：私は当日別の仕事が入ってますから，マスターにやっていただきましょう。
M：な，なにいってるんですか。引き受けるっていったのは琴張さんでしょう。あなた普通にしゃべってればボケになるんだから，やってくださいよ。
護：失礼な。私がいつボケたことをいったというのですか。私は常に物事の論理的構造をとらえ……。
♪カラーン，コローン
伊藤雅彦（以下雅）：いやあ，ここが噂に聞いたZ80's Barかあ。さすがに雰囲気もバーというより薄汚れた事務所というか，意表を突いた演出に一本取られたというさわやかな敗北感が実に心地よい……。
M：あの，もう店はやめて探偵事務所をやってるんですが。
雅：えっ……？
護：あなた，初登場の初ゼリフからいいボケかましますね。その素質を生かして漫才をやってみませんか？
雅：は……？
春：まもるちゃん，それはちょっと無理なんじゃない？
護：いえ，事は急を要するのです。漫才を

やってくれる人間がどうしても必要な状況なのです。
雅：なんだかわかりませんが，人手不足には機械化で対応ってわけで，パソコンに漫才をやらせたらどうです，なんてね。
M：おっ，それは面白いかもしれませんね。
護：とにかく漫才の相方を早く調達したいのです。そのアイデアを採用しましょう。
春：でもプログラムを作らなくちゃいけないんでしょう。まもるちゃんが作るの？
護：いえ，プログラムはもちろんいいだしっぺが作るのです。
雅：えっ？　ぼ，僕が……？
M：さあ，寝袋は用意しますから，泊まり込みでお願いしますよ。

パソコン師匠は学習する

（翌朝）
雅：Ｚzz……。
M：ありゃ，キーボードを枕にして寝ちゃってるよ，この人は。もしもし，起きてください。
雅：…ん……あ？　はっ，い，いかん。寝てしまっ……あー，キーボードによだれが。
M：どうです，できました？
雅：はい，なんとか。
護：S-OS用ですね。
雅：ええ，SLANGで書きました。NEWファイル入出力ライブラリを追加したSLANGコンパイラでコンパイルすると，実行プログラムができます。
M：文字列データに漢字を使ってるようですが。
雅：漢字の出ない機種ではデータをカナに直せば，問題なく動きます。
春：どういうものなの，漫才プログラムって。
雅：えーとですね，漫才ってものは，前フリ・ボケ・ツッコミを繰り返しているもんだと考えたんですよ。それで，前フリの文を入力するとコンピュータがボケを返してきて，それに対するツッコミの文を入力する，という処理を繰り返すプログラムを作ったんです。会話プログラムの変種という感じですね。
M：なるほど，パソコンがボケ役を担当してくれるんですね。ちょっとやってみてくれませんか。
雅：はいはい。（カタカタッ……）
M＆護＆春：……!?
雅：こんな具合です。
M：こんな具合って，ボケがまるっきりトンチンカンじゃないですか。
雅：そりゃそうですよ，まだ学習させてませんから。
春：えっ，学習するの？　すごい，ニューロコンピュータみたいな？
護：まさか，このマヌケ顔の男がニューラルネットを理解しているとは，とても思われません。
雅：ま，そのとおりですが……学習といっても，日本語FEPの「学習機能」と同じレベルの言葉遣いと思ってください。学習情報はmemfileという配列に入ってまして，中身は図1のようなイメージでとらえるとわかりやすいでしょう。
M：主語とか述語っていうのはなんですか？
雅：前フリの文は必ず“A is B”の形になっていなければならないんですよ。
護：“夜空は美しい”というようにですね。
雅：そうです。それで，使える主語と述語はあらかじめ決まっていて（表1），さらにそれぞれの述語について，何種類かのボケが用意されているんです。
M：前フリで主語＋述語を入力すると，その述語に対して用意されているボケのなかからひとつ選んで返答してくるっていう仕組みですね。
雅：はい。図1についてる○の意味なんですが，「そのボケが主語＋述語に対する正しいボケである可能性がある」ということを示しています。たとえば，“君かっこええ”という前フリがきた場合，それに対するボケとしては，“当たり前……”のところにしか○がついていませんから，それをボケとして返すわけです。○がついているところが2カ所以上あったら，どれかひとつを選んで返答します。
春：意外と単純なのね。
護：学習はどのように行われるのですか。つまりmemfileの内容の更新方法についてですが。
雅：ええ，コンピュータはmemfileを参照して，正しい可能性のあるボケを返すわけですが，本当に正しいボケなのかどうかはコンピュータには判断できません。そこで，ボケのあとのツッコミの入力のところで，ボケが正しかったかどうかを人間が判定してやるのです。
M：判定というと？
雅：ボケがおかしかったら，ツッコミ文の入力のときに，最初に“？”マークをつけるというだけです。“？君なにいうてんの”とかね。“？”マークをつけなければ，ボケが正しかったということになります。
春：その判定によってmemfileを書き換えるのね。
雅：そうです。正しいという判定なら，そのボケのところにだけ○をつけて，ほかのボケのところの○は消します。こうすると，今度同じ前フリがきたときに，前回と同じ正しいボケを返すことができるわけです。
M：ボケが間違いという判定だったら，そのボケの所の○を消せば，次回からは選択肢から外れますからＯＫですね。
護：しかし，もしすべてのボケに対して正しくないという判定を下したらどうなるんです？　ボケの選択肢がなくなって，今度同じ前フリが入力されたときになんの返答もできなくなってしまうのでは。
雅：このプログラムは，どんな前フリを入力しても，それにマッチするボケが必ずひとつあるという前提で作ってありますから，普通はすべてのボケが正しくないと判定されることはないと思ってください。ただし，ツッコミ入力のミスなどのために，すべてのボケが否定される可能性はあるわけですよ

表1　登録語一覧

主語	述語
・君　きみ　お前　おまえ　あなた ・僕　ぼく　俺 おれ 私 わたし わし ・男 ・女 ・人 ・お父さん 父さん 親父 おやじ おとん ・お母さん　母さん　おかん ・お兄さん　兄さん　兄貴 ・お姉さん　姉さん　姉貴 ・弟 ・妹 ※主語のうち，“お父さん”～“妹”は，“君の××”，“僕の××”という表現が可能	・かっこええ　かっこいい ・かわいい　可愛い ・きれい　奇麗　綺麗 ・不細工 ・デブ　ブタ ・元気 ・元気ない

図1　MEMファイルの内容

述語：ボケ／主語		君	僕	男	女	人	父	……
かっこええ	当たり前……	○		○		○		
	何言うて……		○			○	○	
	あほ，君……			○		○	○	
	あほな，……			○		○		
	女に「か……				○	○		
かわいい	嘘つけ，……	○	○				○	
	君どんな……	○	○		○	○	○	
	手ェ出す……	○	○			○	○	
	⋮							

ね。そういう場合は，たったいま否定されたボケ以外のボケは選択肢として復活させる，○をつけ直すという処理をしています。

護：なるほど，それなら以前に間違えて否定してしまったボケが，また選択肢に入ってくることになりますね。

雅：はい。具体的にはプログラムの677～689行を見てもらえればわかると思います。ここでmemfileの更新をやってますから。

M：図1でいうところの，○があるとかないとかっていうのは，実際にはビットが立っているかどうかで表されているんですね。

護：ひとつの主語＋述語について2バイトの領域がとってあるということは，選択肢として用意できるボケは，述語ひとつ当たり最大16個ということですか。

雅：そういうことです。

推論してみる

M：でもアレですね，要するに用意してあるボケを片っ端から表示してみて，正しかったらそれを覚えておくだけなんでしょ。もうちょっと知的な処理をやってるかと思ったんですが。

雅：いえいえ，ボケを「片っ端から」表示するなんてことはしてませんよ。選択可能なボケが2つ以上あって，どのようにしてそのなかのひとつを選ぶかというときに，最も正しそうなのを推論する……といってしまうと言葉遣いがふしだらになってしまうのかもしれませんが……，まあそういうことをやっています。

春：へえ，どんなふうに？

雅：まず，23種類の主語をグループ分けするための条件をいくつか作っておきます。それを図示したのが図2になります。

護：ふむ，たとえば“家族か”という条件なら，“お父さん”とか“お母さん”というグループと，“君”とか“人”というグループに分けられるということですね。

M：階層構造になっているようですが？

雅：これは，上位の条件を満たすグループをさらに分類するときだけ，その下位の条件を使うことができるという意味です。たとえば，“家族か”の条件のYes側の枝の下に“親か”という条件がありますが，この“親か”という条件は，家族を表す主語をさらに親と兄弟に分類するために作ってあるんです。

護：なるほど，だから親でも兄弟でもない主語に対して“親か”という条件で分類するのはナンセンスだと，したがって“家族か”という条件でYesになったグループにだけ“親か”という条件を適用することができると，そういうことですね。

雅：そのとおりです。

春：さっすがまもるちゃん，頭の回転が速いんだからー。

雅：……この女は何者……？　まっ，そういうふうにグループ分け条件を作っておいて，グループ内の主語同士で同じボケが選択可能になるまで，グループ分けしていくんです。

M：はあ？

雅：具体例で説明しましょう。話を簡単にするために，使える主語は兄・姉・弟・妹の4つ，述語は仮に述語Aとおいて，述語Aに用意されているボケがボケ1とボケ2の2つ，グループ分け条件は“年上か”“男か”の2つとしましょう（図3）。

M：この例だと，“妹は述語Aだ”という前フリが来たときに，ボケ1かボケ2を選ばなくてはならないんですね。

春：兄と弟はボケ1で，姉はボケ2になっているから，妹は姉と同じボケ2が正しいんじゃないの？

護：つまり“男か”というグループ分け条件で分ければいいということですね。

雅：そのあたりの処理を具体的に説明しますと，まずボケ1が正しいと仮定してグループ分けすると，どうしても3グループに分かれてしまいます(図4－a)。“年上か”という条件で分類すると，兄と姉が1グループになりますが，兄はボケ1，姉はボケ2しか選択できませんから，共通のボケがないわけです。

春：それで今度は“男か”という条件を使って，兄と姉を分けるのね。

M：兄，姉，弟&妹の3グループということですか。

雅：“男か”という条件を最初に使っても，今度は姉と妹が同じグループになります。妹はボケ1と仮定していますから，ボケ2の姉とは別グループにしなければならず，結局3グループに分かれます。

M：なるほどね。妹がボケ2だと仮定した

図2　グループ分け条件

図3　例題の条件

		兄	姉	弟	妹
述語A	ボケ1	○		○	○
	ボケ2		○		○

年上か　Y (兄・姉)　N (弟・妹)
男か　Y (兄・弟)　N (姉・妹)

図4-a　妹＝ボケ1と仮定した場合

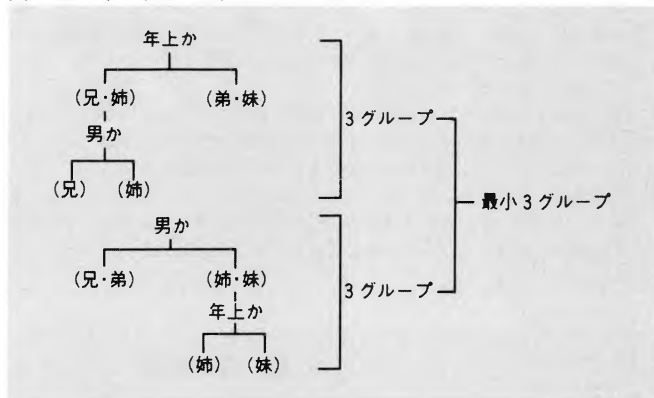
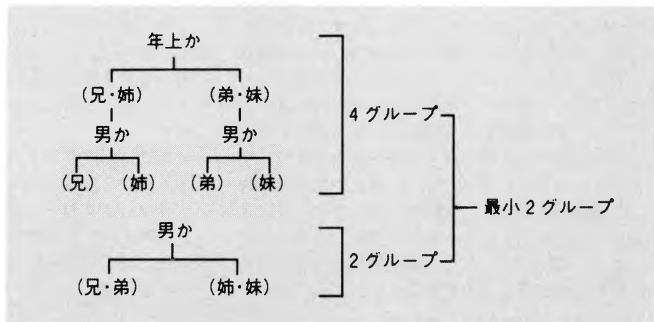
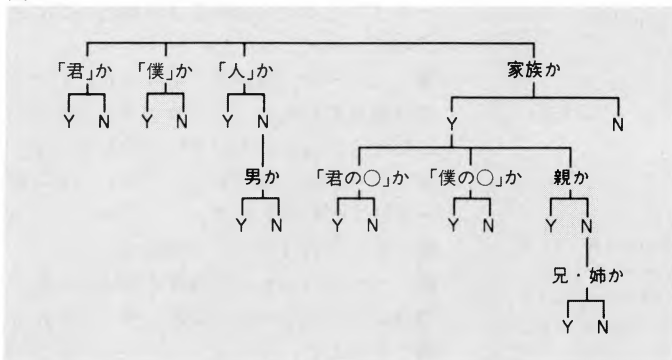
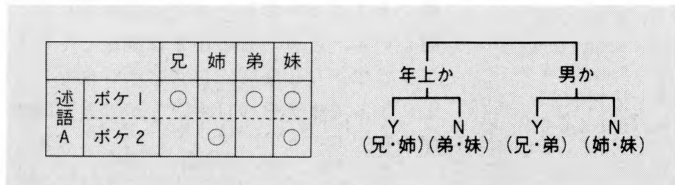



図4-b　妹＝ボケ2と仮定した場合

場合はどうなるでしょう?

雅:“年上か”という条件で先に分けると4グループになってしまいますが,“男か”という条件で先に分けると,それだけで分類が完了して2グループということになります(図4－b)。

護:ボケ2が正しいとしたほうが少ないグループにまとめることができる,そのほうが自然ではないか,ということでボケ2を選ぶ仕組みですね。

雅:そういうことです。プログラムでは599行目からのgroups()関数で,最小グループ数を数えています。関数の戻り値は(グループ数－1)になってますが。

M:groups()関数の中でgroups()関数を使っています。再帰処理ですね。

護:仮引数のmaxはなんなのですか?

雅:これは高速化のためのものなんです。グループ分けというのは,条件をどう組み合わせるかによって何通りでも分け方があって,最小のグループ数を出すためにはそれらを全部試してみなければいけないわけです。

M:それをやっていると遅くなってしまうんですね?

雅:ええ,前フリを入力してから10秒たってもボケが返ってこないということがすぐに起こってしまいます。そこで,要するに「最小の」グループ数がわかればいいんだから,ある条件でグループ分けをしているときに,そのグループ数がそれ以前のグループ分けでのグループ数より多くなるとわかった時点で,それ以上のグループ分けは省略してしまうんです。

M:やっても無駄というわけですね。

雅:ええ,数えようとしているグループ数が,仮引数maxの値より大きくなることがわかったら,もう数えるのはやめてしまうんです。

護:数えるのをやめてしまうということは,もしgroups()の戻り値がmax に与えた値より大きくなっていたら,その戻り値は正しくグループ数を表していないことになるんですか。

雅:そのとおりです。正確なグループ数がわからなくても,maxより大きいということがわかれば十分だということです。

護:なるほどね。

M:さあて,それじゃ早いとこ学習させておいてくれませんか。依頼主の方がもうすぐ来ますんでね。

雅:はいはい。(カタカタッ……)

春:だんだん漫才らしくなってきたわね(実行例)。

♪カラーン,コローン

依:こんにちは。相方の件はどないなりました?

M:お待ちしておりました。このパソコンと,プログラムの入ったフロッピーを持ってってください。

春:よっ,パソコン漫才師たんじょおー!

依:で,でも僕キー打つの苦手やし,だいいちこれボケをしゃべってくれますの?

M&護&春&雅:……!!

依:しゃべってくれな漫才にならんしな。

護:ご,ご心配なく。こちらの者がキーボード入力とボケの読み上げを担当いたしますので……。

雅:ええーっ,僕が?　そりゃないですよ。僕は徹夜でがんばってプログラムを作ったんだから,ほかの人がやってくださいよ。

M:さっき寝てたじゃないですか,キーボードによだれ垂らして。

雅:そ,それはちょっと居眠り……。

依:つべこべいわんと来てくれまへんか。舞台に立つとなったら練習もせなあかんし。

雅:あいたた,そんなーあ……。(ドタバタ,ガチャン)

春:ちょっとかわいそうだったかしら。

護:いえ,きっと心の中では満足しているはずです。なぜなら,自らピエロになることによってオチをつけることができたのですから。

つづく

実行例

```
マエフリ:今日,　君　元気やね
ボケ　　:そらそうや,元気出さなこんなくだらん漫才やってられへんがな
ツッコミ:おいおい,自分の仕事けなすな
マエフリ:僕は　最近　元気ないねん
ボケ　　:君,元気ないことより金ないことの方が深刻やろ
ツッコミ:やかましわ
マエフリ:でも,　今日　僕　おしゃれしてきたから　かっこええやろ
ボケ　　:何言うてんねん,僕の方がかっこええよ
　　　　 僕の顔見てみ,目もあるし口もあるし,鼻なんか穴が2つも開いてんねんで
ツッコミ:そら誰でも一緒や
マエフリ:僕の　おやじも　ちょっと　かっこええねんで
ボケ　　:当たり前や,僕は全日本ナイス・ガイコンテスト関東地区第4ブロックC組準優勝やで
ツッコミ:?誰も君のこと聞いてへんがな
マエフリ:僕の　おやじが　かっこええ言うてんねん
ボケ　　:あほな,あのおやじ,君の顔そのまましわくちゃにした顔してるやんけ
ツッコミ:あほ,そういう言い方すな
マエフリ:親のこと言うんやったら,君の　おかん　ごっつい　不細工やんけ
ボケ　　:君のおかんよりはましやで
ツッコミ:なんやと
マエフリ:まあ,　確かに　僕の　おかんも　かなり　不細工やけども
ボケ　　:君とそっくりやもん,気の毒やわ
ツッコミ:君に言われたないわ
マエフリ:せやけど,　君とこの　妹は　けっこう　きれいやね
ボケ　　:手ェ出すなよ,君にお兄さんと呼ばれたないからな
ツッコミ:アホなこと言いな,嫁にもらいたくなるほどやないわ
マエフリ:ま,きれい言うたら,田中美佐子,　この　女性が　一番　きれいや思うな
ボケ　　:ほんま,あれよりきれいなんはオードリ・ヘップバーンか僕ぐらいやね
ツッコミ:あほ,もうええ加減しなさい,どうも失礼しましたー
```

使い方の説明

プログラムを起動すると,MEMファイル(学習情報)をロードするか聞いてきます。以前にセーブしてあればロードすることができます。続いて記録をとるか聞いてきます。Yesを選択するとカレントドライブにHITOSHIDOC.Txtというファイルを作り,漫才の内容を記録します。

前フリを入力をする際には,実行例を参考にして1語ずつスペース(半角・全角とも可)で区切って入力してください。最後の語が無条件に述語とみなされ,主語はそれ以外のどこにあってもかまいません。また,“君　おかん”とすると,“君の母”という主語と解釈されます。語の頭が主語や述語の登録語と一致すればマッチしたとみなされますから,“かっこええ”を“かっこええなあ”と入力してもかまいません。

漫才を終わるときはSHIFT＋BREAKで抜けてください。ここでMEMファイルをセーブするか聞いてきます。Yesを選択するとカレントドライブにHITOSHIMEM.Ascというファイルが作られます。

最後に,文字列データを半角カタカナに直して入力する際は,makereply()関数のsubtbl, contblに注意してください。ここのデータはキャラクタコード順に並んでいる必要がありますので,カナに直した場合には順番を入れ替える必要があります。

リスト1

```
  1 //
  2 //  Artificial Joker  "ＨＩＴＯＳＨＩ"
  3 //
  4
  5 const docfn = ["HITOSHIDOC.Txt",0],
  6       memfn = ["HITOSHIMEM.Asc",0],
  7       subs = 23,
  8       cons = 7,
  9       grps = 9,
 10
 11       rstr11 = [
 12         "当たり前や、僕は全日本ナイス・ガイコンテスト関東",
 13         "地区第４ブロックＣ組で準優勝やで",0
 14       ],
 15       rstr12 = [
 16         "何言うてんねん、僕の方がかっこええよ¥n僕の顔見て",
 17         "み、目もあるし口もあるし、鼻なんか穴が２つも開い",
 18         "てんねんで",0
 19       ],
 20       rstr13 = [
 21         "あほ、君と血を分けた男がかっこええわけないやろ",0
 22       ],
 23       rstr14 = [
 24         "あほな、あのおやじ、君の顔そのまましわくちゃにし",
 25         "た顔してるやんけ",0
 26       ],
 27       rstr15 = [
 28         "女に「かっこええ」はないやろ",0
 29       ],
 30       rstr21 = [
 31         "嘘つけ、君がそんな顔やのに#0がかわいいわけないや",
 32         "ろ",0
 33       ],
 34       rstr22 = [
 35         "君どんな目してんねん、コンタクトした上に眼鏡かけ",
 36         "とき",0
 37       ],
 38       rstr23 = [
 39         "手ェ出すなよ、君をお兄さんと呼ぶのは嫌やで",0
 40       ],
 41       rstr24 = [
 42         "手ェ出すなよ、君にお兄さんと呼ばれたないからな",0
 43       ],
 44       rstr25 = [
 45         "君がボケてどないすんねん",0
 46       ],
 47       rstr26 = [
 48         "君はホモか",0
 49       ],
 50       rstr31 = [
 51         "嘘つけ、君がそんな顔やのに#0がきれいなわけないや",
 52         "ろ",0
 53       ],
 54       rstr32 = [
 55         "ほんま、あれよりきれいなんはオードリ・ヘップバー",
 56         "ンか僕ぐらいやね",0
 57       ],
 58       rstr41 = [
 59         "君とそっくりやもん、気の毒やわ",0
 60       ],
 61       rstr42 = [
 62         "君の#0よりはましやで",0
 63       ],
 64       rstr43 = [
 65         "ほんま、ああいうの集めてＰＫＯに派遣したらええの",
 66         "にな¥nどっかのゲリラに殺されても悲しくないやろ",0
 67       ],
 68       rstr51 = [
 69         "あほか、僕は体重５５キロメートルやで",0
 70       ],
 71       rstr52 = [
 72         "後ろから頭１発殴っても、気ィついて振り向くまでに",
 73         "３００メートルぐらい逃げられそうやな",0
 74       ],
 75       rstr61 = [
 76         "そらそうや、元気出さなこんなくだらん漫才やってら",
 77         "れへんがな",0
 78       ],
 79       rstr62 = [
 80         "君、覚醒剤もほどほどにしときや",0
 81       ],
 82       rstr63 = [
 83         "そうなん？　人は死ぬ直前に急に元気になるって言う",
 84         "けどな",0
 85       ],
 86       rstr64 = [
 87         "頭からっぽやからな",0
 88       ],
 89       rstr71 = [
 90         "悩み事があってな、相方かえたいねんけどどうしよう",
 91         "か思て",0
 92       ],
 93       rstr72 = [
 94         "君、元気ないことより金ないことの方が深刻やろ",0
 95       ],
 96       rstr73 = [
 97         "君の家族が元気ないのは、君の家に電気がないからや",0
 98       ],
 99       rstr74 = [
100         "現代人はいろいろな悩みを抱えて生きてるもんやで¥n",
101         "今晩のおかず何にしようとか、お昼までに掃除すませ",
102         "なあかんとか",0
103       ],
104
105       condata1 = [              /* かっこええ */
106         5, word rstr11, word rstr12, word rstr13,
107            word rstr14, word rstr15
108       ],
109       condata2 = [              /* かわいい */
110         6, word rstr21, word rstr22, word rstr23,
111            word rstr24, word rstr25, word rstr26
112       ],
113       condata3 = [              /* きれい */
114         6, word rstr31, word rstr32, word rstr23,
115            word rstr24, word rstr25, word rstr26
116       ],
117       condata4 = [              /* 不細工 */
118         4, word rstr41, word rstr42, word rstr43,
119            word rstr25
120       ],
121       condata5 = [              /* デブ */
122         2, word rstr51, word rstr52
123       ],
124       condata6 = [              /* 元気 */
125         4, word rstr61, word rstr62, word rstr63,
126            word rstr64
127       ],
128       condata7 = [              /* 元気ない */
129         4, word rstr71, word rstr72, word rstr73,
130            word rstr74
131       ],
132
133       group9 = [                /* 兄・姉？ */
134         8, 9, 14, 15, 20, 21, $ff, word 0, word 0
135       ],
136       group8 = [                /* 親？ */
137         6, 7, 12, 13, 18, 19, $ff, word 0, word group9, word 0
138       ],
139       group7 = [                /* 僕の××？ */
140         18, 19, 20, 21, 22, 23, $ff, word 0, word 0
141       ],
142       group6 = [                /* 君の××？ */
143         12, 13, 14, 15, 16, 17, $ff, word 0, word 0
144       ],
145       group5 = [                /* 家族？ */
146         6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
147         20, 21, 22, 23, $ff,
148         word group6, word group7, word group8, word 0, word 0
149       ],
150       group4 = [                /* 男？ */
151         1, 2, 3, 6, 8, 10, 12, 14, 16, 18, 20, 22, $ff,
152         word 0, word 0
153       ],
154       group3 = [                /* 「人」？ */
155         5, $ff, word 0, word group4, word 0
156       ],
157       group2 = [                /* 「僕」？ */
158         2, $ff, word 0, word 0
159       ],
160       group1 = [                /* 「君」？ */
161         1, $ff, word 0, word 0
162       ];
163
164 var   docflg,
165       words,
166       subwdno,
167       subno,
168       conno,
169       replyno,
170       replypos;
171
172 array word memfile[cons-1][subs-1],
173       word condtptr[cons-1] = [
174         word condata1, word condata2, word condata3,
175         word condata4, word condata5, word condata6,
176         word condata7
177       ],
178       byte sentence[255],
179       byte reply[255],
180       word wordptr[9];
181
182
183 main()
184
185   var mfptr, lsw, initdt,
186       i, j;
187
188   begin
189                                /* ＭＥＭファイル準備 */
190     print("ＭＥＭファイルをロードしますか？");
191     lsw = 0;
192     loop [
193       case (inkey(1)) [
194         'Y','y':[
195           print("¥n");
196           if (fopen(1,memfn,0)) [
197             print("ファイルオープンに失敗しました。¥n");
198             print("ロードは行いません。¥n");
199           ] else [
200             if (fseek(1,0,2) != cons*subs*2) [
201               print("ファイルサイズが異常です。¥n");
202               print("ロードは行いません。¥n");
203             ] else [
204               fseek(1,0,0);
205               mfptr = memfile;
206               for i=1 to cons*subs*2 [
207                 mem[mfptr++] = fgetc(1);
208               ]
209               lsw = 1;
210             ]
211             fclose(1);
212           ]
213           exit;
214         ]
215         'N','n','¥n':[
216           print("¥n");
217           exit;
218         ]
219       ]
220     ]
221     if (lsw == 0) [
222       for i=0 to cons-1 [
223         initdt = 0;
224         j = mem[condtptr[i]];
225         repeat [
226           initdt = initdt * 2 + 1;
227         ] until (--j == 0);
228         for j=0 to subs-1 [
229           memfile[i][j] = initdt;
230         ]
231       ]
232     ]
```

```
233                                   /* 記録ファイルオープン */
234     print("記録をとりますか?");
235     loop [
236       case (inkey(1)) [
237         'Y','y':[
238           docflg = 1;
239           exit;
240         ]
241         'N','n','¥n':[
242           docflg = 0;
243           exit;
244         ]
245       ]
246     ]
247     print("¥n");
248     if (docflg) [
249       if (fopen(0,docfn,3)) [
250         print("ファイルオープンに失敗しました。¥n");
251         print("記録はとりません。¥n");
252         docflg = 0;
253       ]
254     ]
255                                   /* 漫才のひととき */
256     applaud();
257     loop [
258       print("［フリ］¥n");
259       if (getl(sentence) == -1)  exit; /* BREAKキーチェック */
260       furu();
261       print("［ボケ］¥n");
262       bokeru();
263       print("［ツッコミ］¥n");
264       if (getl(sentence) == -1)  exit; /* BREAKキーチェック */
265       tsukkomu();
266     ]
267     applaud();
268                                   /* 記録ファイルクローズ */
269     if (docflg) [
270       fputc(0,0);
271       while (fclose(0)) [
272         print("ファイルクローズに失敗しました。¥n");
273         print("リトライしますか?");
274         loop [
275           case (inkey(1)) [
276             'Y','y','¥n': exit;
277             'N','n'     : exit(2);
278           ]
279         ]
280         print("¥n");
281       ]
282     ]
283                                   /* MEMファイルセーブ */
284     print("MEMファイルをセーブしますか?");
285     loop [
286       case (inkey(1)) [
287         'Y','y':[
288           print("¥n");
289           if (fopen(1,memfn,3)) [
290             print("ファイルオープンに失敗しました。¥n");
291             print("セーブは行いません。¥n");
292           ] else [
293             mfptr = memfile;
294             for i=1 to cons*subs*2 [
295               fputc(1,mem[mfptr++]);
296             ]
297             fclose(1);
298           ]
299           exit;
300         ]
301         'N','n','¥n':[
302           print("¥n");
303           exit;
304         ]
305       ]
306     ]
307   end;
308
309
310 applaud()                     /* 拍手 */
311
312   var i;
313
314   begin
315     print("¥n¥n");
316     for i=1 to 8 [
317       print("パチ");
318     ]
319     print("¥n¥n");
320   end;
321
322
323 furu()                        /* フる */
324
325   var wdno, topflg, spcsw, i, j;
326
327   begin
328     if (docflg) [
329       fputc(0,'¥n');
330       fputc(0,$81);             /* "「" */
331       fputc(0,$75);
332     ]
333
334     topflg = 1;
335     wdno = i = 0;
336     loop [
337       case (sentence[i]) [
338         $00   : exit;          /* 文末 */
339         $20   : spcsw = 1;     /* 半角スペース */
340         $81   : if (sentence[i+1] == $40) [
341                   spcsw = 2;  /* 全角スペース */
342                 ] else [
343                   spcsw = 0;
344                 ]
345         others: spcsw = 0;
346       ]
347
348       if (spcsw) [             /* スペース? */
349         repeat [
350           sentence[i++] = 0;
351         ] until (--spcsw == 0);
352         topflg = 1;
353       ] else [
354         if (docflg)  fputc(0,sentence[i]);
355         if (topflg) [          /* 単語の先頭? */
356           if (wdno < 10) [
357             wordptr[wdno++] = sentence + i;
358           ] else [
359             for j=0 to 8 [
360               wordptr[j] = wordptr[j+1];
361             ]
362             wordptr[9] = sentence + i;
363           ]
364           topflg = 0;
365         ]
366         i++;
367       ]
368     ]
369     words = wdno;
370
371     if (docflg) [
372       fputc(0,$81);            /* "」" */
373       fputc(0,$76);
374       fputc(0,'¥n');
375     ]
376   end;
377
378
379 bokeru()                       /* ボケる */
380
381   var i;
382
383   begin
384     replypos = 0;
385     makereply();
386     reply[replypos] = 0;
387     print(msx$(reply),"¥n");
388     if (docflg) [
389       fputc(0,'¥n');
390       fputc(0,$81);            /* "『" */
391       fputc(0,$77);
392       i = 0;
393       while (reply[i]) [
394         if (reply[i] == '¥n') [
395           fputc(0,$81);        /* "』" */
396           fputc(0,$78);
397           fputc(0,'¥n');
398           fputc(0,$81);        /* "『" */
399           fputc(0,$77);
400           i++;
401         ] else [
402           fputc(0,reply[i++]);
403         ]
404       ]
405       fputc(0,$81);            /* "』" */
406       fputc(0,$78);
407       fputc(0,'¥n');
408     ]
409   end;
410
411
412 makereply()
413
414   const subtbl = [
415           "あなた"    , 0, 1, "おかん"    , 0, 7,
416           "おとん"    , 0, 6, "おまえ"    , 0, 1,
417           "おやじ"    , 0, 6, "おれ"      , 0, 2,
418           "お兄さん"  , 0, 8, "お姉さん"  , 0, 9,
419           "お前"      , 0, 1, "お父さん"  , 0, 6,
420           "お母さん"  , 0, 7, "きみ"      , 0, 1,
421           "ぼく"      , 0, 2, "わし"      , 0, 2,
422           "わたし"    , 0, 2, "俺"        , 0, 2,
423           "君"        , 0, 1, "兄さん"    , 0, 8,
424           "兄貴"      , 0, 8, "姉さん"    , 0, 9,
425           "姉貴"      , 0, 9, "私"        , 0, 2,
426           "女"        , 0, 4, "親父"      , 0, 6,
427           "人"        , 0, 5, "男"        , 0, 3,
428           "弟"        , 0,10, "父さん"    , 0, 6,
429           "母さん"    , 0, 7, "僕"        , 0, 2,
430           "妹"        , 0,11, $ff
431         ],
432         contbl = [
433           "かっこいい", 0, 1, "かっこええ", 0, 1,
434           "かわいい"  , 0, 2, "きれい"    , 0, 3,
435           "デブ"      , 0, 5, "ブタ"      , 0, 5,
436           "可愛い"    , 0, 2, "奇麗"      , 0, 3,
437           "元気"      , 0, 6, "元気ない"  , 0, 7,
438           "不細工"    , 0, 4, "綺麗"      , 0, 3, $ff
439         ],
440
441         rstr01 = ["??どないしてん?",0],
442         rstr02 = ["??なんやこの「間」は",0],
443         rstr03 = [
444           "??君がしゃべる番やん、あ・うんの呼吸でいかな!",0
445         ],
446         rstr04 = ["??なんやて?",0],
447         rstr05 = ["??なんやわけわからんわ",0],
448         rstr06 = ["??誰がいな?",0];
449
450   array word replystr1[2] = [
451           word rstr01, word rstr02, word rstr03
452         ],
453         word replystr2[1] = [word rstr04, word rstr05],
454         byte sl[subs] = [
455            1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12,
456           13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,  0
457         ],
458         word gl[4] = [
459           word group1, word group2, word group3, word group5,
460           word 0
461         ];
462
463   var   memdata, gs, bestgs, bestreply, bests,
464         i;
465
466   begin
467     replyno = 0;
468     if (words == 0) [          /* 文が入力されてない? */
```

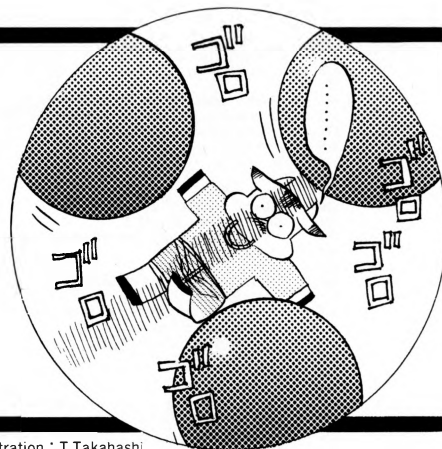

▶いやー，すごいすごい。ちゃんとモーフィングを実現している。写真を見た限りでは，ほぼ完璧だと思います。　吉野　貴宣(18)広島県

```
469       wrreply(replystr1[rnd(3)]);
470       return;
471     ]
472     conno = search(contbl,wordptr[words-1]);
473     if (conno == 0) [          /* 補語が未登録？ */
474       wrreply(replystr2[rnd(2)]);
475       return;
476     ]
477
478     subno = 0;
479     subwdno = words - 2;
480     while (subwdno != -1) [
481       if (subno = search(subtbl,wordptr[subwdno]))  exit;
482       subwdno--;
483     ]
484     if ((subno >= 6) and (subno <= 11)) [  /* 主語が家族？ */
485       if (subwdno >= 1) [
486         case (search(subtbl,wordptr[subwdno-1])) [
487           1: subno = subno + 6;
488           2: subno = subno + 12;
489         ]
490       ]
491     ]
492     if (subno == 0) [          /* 主語がない？ */
493       wrreply(rstr06);
494       return;
495     ]
496
497     memdata = memfile[conno-1][subno-1];
498     bestgs = 65000;
499     for i=0 to 15 [
500       if (bit(memdata,i)) [
501         memfile[conno-1][subno-1] = set(0,i);
502         gs = groups(sl,gl,bestgs);
503         if (gs < bestgs) [
504           bestgs = gs;
505           bestreply = i;
506           bests = 1;
507         ] else [
508           if (gs == bestgs) [
509             if (rnd(++bests) == 0)  bestreply = i;
510           ]
511         ]
512       ]
513     ]
514     memfile[conno-1][subno-1] = memdata;
515     wrreply(memw[condtptr[conno-1]+1+bestreply*2]);
516     replyno = bestreply + 1;
517   end;
518
519
520 search(tblptr,byte wd[])  /* テーブルサーチ */
521
522   var ans, i;
523
524   begin
525     ans = 0;
526     loop [
527       i = 0;
528       while (wd[i] == mem[tblptr]) [
529         if (mem[++tblptr] == 0) [
530           ans = mem[tblptr+1];  /* サーチ成功 */
531           exit;
532         ]
533         i++;
534       ]
535       if (wd[i] < mem[tblptr])  exit;  /* サーチ終了 */
536       while (mem[tblptr++]) []  /* 不一致データをスキップ */
537       tblptr++;
538     ]
539   end(ans);
540
541
542 wrreply(byte string[])
543
544   const sstr01 = ["僕",0],
545         sstr02 = ["君",0],
546         sstr03 = ["男",0],
547         sstr04 = ["女",0],
548         sstr05 = ["人",0],
549         sstr06 = ["おやじ",0],
550         sstr07 = ["おかん",0],
551         sstr08 = ["兄貴",0],
552         sstr09 = ["姉貴",0],
553         sstr10 = ["弟",0],
554         sstr11 = ["妹",0];
555
556   array word substr[subs-1] = [
557           word sstr01, word sstr02, word sstr03, word sstr04,
558           word sstr05, word sstr06, word sstr07, word sstr08,
559           word sstr09, word sstr10, word sstr11, word sstr06,
560           word sstr07, word sstr08, word sstr09, word sstr10,
561           word sstr11, word sstr06, word sstr07, word sstr08,
562           word sstr09, word sstr10, word sstr11
563         ];
564
565   var   substrptr,
566         i;
567
568   begin
569     i = 0;
570     loop [
571       case (string[i]) [
572         0     : exit;
573         '#'   :[
574                 case (string[++i]) [
575                   0  : exit;
576                   '0':[
577                         substrptr = substr[subno-1];
578                         while (mem[substrptr]) [
579                           wrchr(mem[substrptr++]);
580                         ]
581                   ]
582                 ]
583                 i++;
584         ]
585         others: wrchr(string[i++]);
586       ]
587     ]
588   end;
589
590
591 wrchr(chr)
592   begin
593     if (replypos < 255) [
594       reply[replypos++] = chr;
595     ]
596   end;
597
598
599 groups(byte sublist[],word grplist[],max)
600   begin
601     array byte sl1[subs], byte sl2[subs],
602           word gl[grps];
603
604     var   grpptr, ans, gs,
605           i, j, k, l, m;
606
607     j = $ffff;
608     i = 0;
609     while (sublist[i]) [
610       j = j and memfile[conno-1][sublist[i++]-1];
611     ]
612     if (j)  return(0);         /* 共通のボケがある */
613
614     ans = max + 1;
615     j = 0;
616     while (grpptr = grplist[j]) [
617       if (ans == 1)  return(1);
618       k = l = m = 0;
619       while (sublist[k]) [
620         if (sublist[k] > mem[grpptr]) [
621           grpptr++;
622         ] else [
623           if (sublist[k] < mem[grpptr]) [
624             sl2[m++] = sublist[k++];
625           ] else [
626             sl1[l++] = sublist[k++];
627             grpptr++;
628           ]
629         ]
630       ]
631       while (mem[grpptr++] != $ff) []
632
633       if ((l > 0) and (m > 0)) [
634         sl1[l] = sl2[m] = 0;
635         k = l = 0;
636         while (grplist[k]) [
637           if (j != k)  gl[l++] = grplist[k];
638           k++;
639         ]
640         m = 1;
641         while (memw[grpptr]) [
642           gl[m++] = memw[grpptr];
643           grpptr = grpptr + 2;
644         ]
645         gl[m] = 0;
646         gs = 1 + groups(sl1,gl,ans-2);
647         if (gs < ans) [
648           grpptr = grpptr + 2;
649           m = 1;
650           while (memw[grpptr]) [
651             gl[m++] = memw[grpptr];
652             grpptr = grpptr + 2;
653           ]
654           gl[m] = 0;
655           gs = gs + groups(sl2,gl,ans-gs-1);
656           if (gs < ans)  ans = gs;
657         ]
658       ]
659       j++;
660     ]
661     if (ans > max)  ans = i - 1;  /* グループ分け失敗 */
662   end(ans);
663
664
665 tsukkomu()                    /* ツッコむ */
666
667   var spt,
668       i, j;
669
670   begin
671     spt = 0;
672     case (sentence[0]) [
673       '?': spt = 1;                                /* 半角疑問符 */
674       $81: if (sentence[1] == $48)  spt = 2;  /* 全角疑問符 */
675     ]
676     if (replyno) [
677       if (spt) [
678         i = memfile[conno-1][subno-1];
679         if (reset(i,replyno-1) == 0) [
680           i = 0;
681           j = mem[condtptr[conno-1]];
682           repeat [
683             i = i * 2 + 1;
684           ] until (--j == 0);
685         ]
686         memfile[conno-1][subno-1] = reset(i,replyno-1);
687       ] else [
688         memfile[conno-1][subno-1] = set(0,replyno-1);
689       ]
690     ]
691
692     if (docflg) [
693       fputc(0,'¥n');
694       fputc(0,$81);            /* "「" */
695       fputc(0,$75);
696       while (sentence[spt]) [
697         fputc(0,sentence[spt++]);
698       ]
699       fputc(0,$81);            /* "」" */
700       fputc(0,$76);
701       fputc(0,'¥n');
702     ]
703   end;
```

▶キーボードの足って，どこでどれぐらいで売ってんだろう？　鷲見　健二(18)大阪府

いけいけごぉごぉ3D

Komura Satoshi 古村 聡

今月のショートプロはゲームが2本，ツールが1本。ゲームのリストはとっても短い。ツールはアセンブラでチョッピリ長め。すぐに打ち込んで楽しんでね。おっと忘れちゃいけない，ぱーてぃハンズの格闘ゲームが今回で完成。バンザーイ。

illustration：T.Takahashi

どうもおこんちわ，(で)でございます。

バーチャルリアリティじゃないけど，最近は「本物じゃないのに本物に見せる」という技術が本当に流行ってますね。といえばCGとか「ジュラシックパーク」の話と思うかもしれないけどそうじゃなくて。

このあいだ近所のコンビニで買った明治の「純情りんご」という100円のパックのジュースを飲んだんですよ。これがパッケージに「すりりんごのような飲みごこち」と書いてあるとおり，本当にりんごをすりおろしたようなザラザラっていう感触がするんです。で，本当にりんごのすりおろしたものが入っているんならまだ納得がいくんだけど，なんとこれが果汁3%。果肉なんかこれっぽっちも入っていないというすごいモドキドリンクだったんですね。

原材料を見ると「ゲル化剤(ペクチン)」と書いてあるんで，よくわかんないけどきっとなんか薬で無理矢理ザラザラにしてるんでしょう。いや，なんともすごいんですね，最近の技術ってやつは。

最近のつぶつぶオレンジとかの果肉の入ったジュースってきっと，「本物の果実をしぼったジュースに近づけたい」っていう希望から出てきたものだと思うんですね。でも，安く作るには偽物でそれっぽく作らなきゃならないから，この「純情りんご」みたいな「より本物っぽく見せるために高度な技を使って作った，手のこんだ偽物」なんて物が出てきちゃうんですね。でも，こういう物が出てくると，本物と偽物の差ってなんだ，って思っちゃうんですよ。あんまり偽物がよくできてきちゃうと偽物と本物って区別する必要がなくなってしまって，偽物が本物らしくある必要がなくなってしまうわけで，それじゃあ，偽物を作る意義ってなんなのかなと……。

ま，いいんですけどね。

今月も3Dでごぉごぉ

では今月のプログラムにいきましょう。今月の1本目はX－BASIC用の3Dゲーム，AVOID.BASです。どうぞっ！

AVOID.BAS for X68000/030
(要X-BASIC，ジョイスティック)
三重県　平井栄治

8月号，高木さん作のSCROLL3Dに続いての揺れる揺れる3Dゲームなんでありますね。思わず体がジョイスティックと一緒に揺れちゃいます～。

このプログラムはX-BASIC用のゲームです。BASICを起動して，リスト1を入力したらRUNでゲームが始められます。ジョイスティック専用ゲームなので，ジョイスティックを本体につなげるのを忘れないでくださいね(こらこら，「ジョイパッドじゃあかんの？」とかいわないよーに。どっちでもいいんだから，つながれば)。

さーて，ゲームが始まるとちょっとの間，待ち時間があったあと，ゲーム画面が表示されます。ジョイスティックのボタンを押してゲームスタート！　するするするっと画面の奥のほうから赤い玉が流れてきます。この玉にぶつからないないように緑の○をジョイスティックを左右に動かしてよけてください。ぶつかるまでの時間がそのままスコアになります。

AVOID.BAS

さてさてこのゲームなんですが，8月号のSCROLL3Dがドットスクロールゲームの3D版なら，こっちはスクロールで流れてくる石をよけろ！　ゲーム3D版なんでありますね。プログラム的にもSCROLL3Dと同じようにパレットアニメーションを使ったプログラムなのであります。パレットアニメーションって何かというと……8月号を見てくださいね。

さすが，絵が素早く描けるパレットアニメーションを使ってるだけあって，BASICにしてはなかなかのスピードで遊べますね。8月号のSCROLL3Dのように10MHz以外のマシンだと速すぎるということはないですけど，10MHzでも十分遊べるスピードです。

さーて，ドットスクロールゲーム，スクロールで流れてくる物をよけろモノ(どうにかならんのかこの長い名前は……)と，ショートプロの両王道ゲームの3Dゲームが続いてるけど次は何が出てくるかな⁉　ライトサイクル3Dかそれともパズルゲームの3Dかなっ？　わくわくっ(おいおいって)。マインスイーパーだったら……ちょっとこわいかもしれない……。

叩く叩くハエは逃げる

では，次のプログラムにいきましょう。2本目は東京都の水野さん作，X1turbo用ゲームでFLY.BASです。どうぞっ！

FLY.BAS for X1turbo
(turboBASIC用，要マウス)
東京都　水野浩一

夏。あつーい夏をさらにうっとうしくしてくれる生き物。食べ物にまとわりつき，その子供は腐ったすいかの皮にうぞうぞとわき，音，ストレス，伝染病，あらゆるイヤな物を運んでくる，あの，「ハエ」の動きを忠実に再現したゲームです。

このプログラムはturboBASIC用のプログラムです。まずX1turboのRESOLUTIONスイッチをHIGHにして高解像度の状態でturboBASICを立ち上げてください。そうそう，このゲームはマウス専用ですからマウスも用意しておいてくださいね。それからリスト2を入力してRUNします。

画面上にテキストで“O”と“a”が現れます。“O”がハエタタキでマウスを動かすことで上下左右に移動することができ，左クリックで叩くことができます。

で，“a”がにっくきハエ。こいつがチョコマカチョコマカと動き回り，近くにハエタタキがくると逃げます。ハエタタキをうまく使ってこのハエを叩いていってください。

ハエを叩き損ねると－10点でtimeが減ります。timeが0になるとゲームオーバーです。

おおおおお，うっとうしいぞ～っ！　なんてうっとうしい動きをするハエなんでしょ。本物より本物っぽいハエの動きをするこのプログラムハエです。うううううう，いやだなー。

もはやちょっと季節が過ぎちゃったけど(投稿自体はかなり早く来てたんだけど，最近ちょっと掲載が遅れぎみなんです。ゴメンね)，プログラムのサイズも小さいし，とってもよろしいんではないかと思います。しかしこれ，X1じゃなくてX68000でキャラクターをスプライトで描いてPCMでハエの羽音の効果音つきだったりしたら……ん～，想像するだけでイヤだ……。X1でよかったかもしれないな，ふと思ってしまう私なのでありました。いやはや，これの3D版とか……ほとんど夏休み納涼怪奇の世界へようこそになってしまいまんがな(まだこだわっている)。

最後はすいすいディレクトリだ!

でもって，今月の最後のプログラム。X68000用，ディレクトリをすいすい移動，XCD.Sです。どうぞっ！

XCD.S for X68000/030

(要アセンブラ，リンカ)

宮崎県　岡田晃一

このプログラムはカーソルキーを使ってワンタッチでディレクトリを移動するためのプログラムです。

このプログラムのリストはアセンブラのソースの形になっています。実行ファイル形式にするにはアセンブラとリンカ，それにマクロファイルDOSCALL.MAC，IOCSCALL.MACが必要になります。

リスト1　AVOID.BAS

```
 10      o=1:screen 0,1,1,1: m_init():dim r(8),t(8),v(8),x(8)
 20        dim char z(1818):sp_init():dim s(8),u(8),w(8),y(8)
 30        for n=0 to  8:v(n)=14*n+88:   palet(n,n*  6+  0  )
 40        r(n)= 9*n+108:w(n)=16*n+80:   palet(0,         0  )
 50        s(n)=10*n+104:x(n)=20*n+64:sp_color(n,n*192+192,1)
 60        t(n)=11*n+100:y(n)=24*n+48:sp_color(0,         0,1)
 70        u(n)=12*n+ 96:next:sp_on():sp_color(9,     51201,1)
 80            m_alloc(1,818):m_assign(1,1):m_trk(1,"t181r8")
 90            sp_disp(1):m= 4:a():m= 4:b():m= 5:a():m= 5:b()
100                       m= 6:b():m= 7:b():m= 9:c():m=11:c()
110                            circle(15,15,m,o):d():k= 0:o=8
120                       m=96:e():m=80:e():m=64:e():m=56:e()
130                       m=48:e():m=44:e():m=40:e():m=36:e()
140 locate  5, 0:print"High score  ":for n=0 to 7:z(n)=0:next
150 locate  5, 1:print"     Score  ":m=0:n=4 :f():        q=0
160 locate 10,15:print"AVOID   .BAS"+chr$(11):h():  while q=0
170 locate 13,14:print      "Press."+chr$( 7):h():        q=0
180            while strig(1)=0:h():if q<>0 then{break}else{}
190                        endwhile:if q<>0 then{break}else{}
200 locate 13,14:print      "        ":for n=0 to 7:z(n)=0:next
210 locate 18, 1:print            "  ":m=0:n=4 :f():        q=0
220      p=0:m_play():while q=0:while p<>0:while m_stat()<>0
230              p=stick(1):if p=0 then{break}else{}:endwhile
240               break:endwhile:while m_stat()<>0:p=stick(1)
250  if p<>0 then{j=p:    switch p:case 1:case 4:case 7:n=n-1
260  if n=-1 then{n=0}else{}:break:case 3:case 6:case 9:n=n+1
270  if n= 8 then{n=7}else{}:break: endswitch:break}else{j=0}
280 endwhile:while m_stat()<>0:endwhile:m_play():z(o)=rnd()*8
290       m=m+1:f():h():if j<>  0 then{ p=stick(1)}else{p=0}
300                      if m= 999 then{break}else{}
310                      if n=z(o) then{break}else{}:endwhile
320            endwhile:print      chr$(7)+chr$(11):end
330 func a():circle(m  ,m  ,m,o): paint( m, m,o):d():endfunc
340 func b():circle(m  ,m+1,m,o):circle(m+1,m  ,m,o)
350         circle(m+1,m+1,m,o):                     a():endfunc
360 func c():circle( 15, 15,m,o): paint(15,15,o):d():endfunc
370 func d():k=0:for p=0 to 15:for q=0 to 15:z(k)=point(q,p)
380 k=k+1:next:next:sp_def(n,z,1):o=o+1:n=n+4:wipe():endfunc
390 func e():  p=127+m:q=128-m:fill(q,q,p,p,o):o=o-1:endfunc
400 func f():     if k<m then{k=m}else{}:locate 16,0:print k
410 sp_set(17,r(z(o)),171, &H109,3):g() :locate 16,1:print m
420 sp_set(16,s(z(o)),174, &H10D,3):g()
430 sp_set(15,t(z(o)),177, &H111,3):g()
440 sp_set(14,u(z(o)),180, &H115,3):g()
450 sp_set(13,v(z(o)),186, &H119,3):g()
460 sp_set(12,w(z(o)),192, &H11D,3):g() : p=x(z(o))-6:q=p+16
470 sp_set(11,       p,198, &H121,3):sp_set(9,p,214,&H8121,3)
480 sp_set(10,       q,198,&H4121,3):sp_set(8,q,214,&HC121,3)
490                                  g() : p=y(z(o))-4:q=p+16
500 sp_set( 7,       p,212, &H125,3):sp_set(5,p,228,&H8125,3)
510 sp_set( 6,       q,212,&H4125,3):sp_set(4,q,228,&HC125,3)
520                                         p=y(   n)-4:q=p+16
530 sp_set( 3,       p,212, &H129,3):sp_set(1,p,228,&H8129,3)
540 sp_set( 2,       q,212,&H4129,3):sp_set(0,q,228,&HC129,3)
550                                                   endfunc
560 func g():            o=o+1:if o=8 then{o=0}else{}:endfunc
570 func h():if inkey$(0)=chr$(27) then{q=1818}else{
580                q=0:while inkey$(0)<>"":endwhile}:endfunc
590 /* AVOID             BAS         3695  93-06-19  12:00:00
```

リスト2　FLY.BAS

```
10 MOUSE 1,40,15:MOUSE 2,0,3:MOUSE 2,1,6:MOUSE 3,0,0,79,23:KLIST
 0:CONSOLE 0,25
20 CLS:DEFINT a-z:s=0:x=RND*79:y=RND*23:t=1000
30 c=MOUSE (0):d=MOUSE (1)
40 IF a<>c OR b<>d THEN LOCATE a,b:PRINT " ";
50 a=c:b=d:GOSUB 300:LOCATE a,b
60 IF a=x AND b=y THEN PRINT "@";: ELSE PRINT "O";:LOCATE x,y:PR
INT "a";
70 IF MOUSE (2,1) GOSUB 210 ELSE m=0
80 IF z>0 THEN 260
90 LOCATE x,y:PRINT " ";
100 x=x+RND*8-4:x=x + (x>79)*(2*x-158) + (x<0)*(2*x)
110 y=y+RND*6-3:y=y + (y>23)*(2*y-46) + (y<0)*(2*y)
120 IF RND<.05 THEN z=RND*150
130 IF s<-990 THEN 150
140 t=t-1:IF t>-1 THEN 30
150 t=0:GOSUB 300:LOCATE 25,7:PRINT "Game Over !!":LOCATE 30,12:
PRINT "Replay ?"
160 LOCATE 27,9:PRINT "Score ---",s
170 i$=INKEY$
180 IF i$="y" OR i$="Y" THEN RUN
190 IF i$<>"n" AND i$<>"N" THEN 170
200 KLIST 1:CONSOLE 0,24:CLS:END
210 IF m RETURN
220 m=-1:z=0
230 BEEP:LOCATE a,b:PRINT "*";
240 IF a<>x OR b<>y THEN s=s-10*INT ((z+10)/10):t=t-RND*30:RETUR
N
250 PLAY "c":s=s+INT (z/5)*30 - (z=0)*2000:x=RND*79:y=RND*23:RET
URN
260 z=z-1:l=(a-x)^2+(b-y)^2
270 z=z + (l=1)*20 + (l<10)*10 + (l<26)*8 + (l<50)*5 + (l<82)*1
280 IF z<0 THEN z=0
290 GOTO 140
300 LOCATE 10,24:PRINT USING "SCORE :   #####     TIME :   ####"
,s,t;:RETURN
```

例によってエディタでソースリストを入力し，それからアセンブル，リンクという手順を踏んで実行ファイルを作ってください。福袋やXCに入っているアセンブラとリンカを使うのであれば，

A＞AS XCD.S
A＞LK XCD.O

で作ることができます(ただし，福袋にはマクロファイルは入っていませんので注意してください)。

さて，無事にアセンブルが済んだら使い方について説明します。

このプログラムの使い方はただひたすら

A＞XCD

のみです。オプションはありません。ヘルプも出力しません(何かオプションをつけようとするとこのプログラムの名前だけ表示して終了します)。

プログラムを実行するとカレントディレクトリから見えるディレクトリ(親ディレクトリを含みます)の一覧を表示します。

そこで行きたいディレクトリをカーソルキーで選んでください。そして，リターンキーを押すとそのディレクトリに移動します。ESCキーで移動しないで終了します。

はいっ，このショートプロでもたびたび登場しているディレクトリ移動モノのプログラムですね。えーっと，最近だと'92年の12月号で登場したCDS.Cっていうディレクトリのリストを作ってワンタッチジャンプするプログラムっていうプログラムがありますね。

このXCD.Sは300行近くとちょっとショートプロとしては大きめのプログラムになっていますね。まぁ，アセンブラだからしかたがないといえばそのとおりなんですけど……。アセンブラにはアセンブラの道があると思うのですよね。たとえば，アセンブラってCに比べて割り込み処理を書くのが楽でしょ。だから，コピーキーの割り込みベクタを乗っ取って，コピーキー一発でこのプログラムが立ち上がるとか……長くなっちゃうから大変かもしれないけど(もちろん「Cに不可能はない！」ってんでCでそういうプログラムを書いてくれてもかまわないんですけどね)。

そうそう。このプログラムはキー入力周りをオリジナルではDOSの１文字入力からIOCSのリアルタイムキー入力に変更しています。というのも，DOSコールの１文字入力はKEY.Xなどによって細工される可能性があるので，カーソルキーを取り込むつもりが，他の文字列が入力されてあらぬ動作をする可能性があるので，こういうプログラムのときには向かないんですね。皆さんも自分でプログラムを作るときには十分気をつけて，どちらの入力を使うか選んでくださいね。

ま，言語にしてもＣ言語，アセンブラかそれともBASIC,あるいはプログラムにしてもIOCS/DOSコール，直接ハードを叩くか，それともＣのライブラリなど，使えるものがたくさんあって悩むものですけど，それぞれの物にそれぞれ得意分野がありますんで，それを生かしたプログラミングをこころがけるってことがやっぱり大事なのですよね。それこそが美しくて，使いやすい，いいプログラムを生むのです。人生精進じゃ。

と，結論が出たところで今月はおしまい。また来月お会いしましょう。それでは，あでゅ〜。

動かないよと思う前に(12)

★リストを入れたけどキャラが……

「ぱーてぃハンズ」のリストを入れたけどキャラクターがぐちゃぐちゃになってるよ，という人がいませんか？

この「ぱーてぃハンズ」のリストは，ゲームのプログラム部分だけで，キャラクターを作るスプライトパターンやスプライトのパレットデータは入っていません。

ゲームをするには，'92年12月号のショートプロぱーてぃのリスト４と今月号の付録ディスクについているMAC.XとLHA.Xが必要です。

まず，MAC.Xを使ってリスト４を打ち込み，PAT2.LZHという名前でセーブします。それからLHAでPAT2.LZHを展開してPAT2.BASというプログラムをRUNします。これでパターンとパレットの設定ができました。あとはハンズのプログラムをRUNすれば，ゲームをすることができます。

MAC.Xなどのくわしい使い方については，'92年12月号を参照してください。

リスト3 XCD.S

```
 1: ****************************************
 2: * X68k XCD ver1.00 Copyright 1993 K.O  *
 3: ****************************************
 4:         .include        DOSCALL.MAC
 5:         .include        IOCSCALL.MAC
 6: MAX             equ     50      *最大検索数
 7: LEN             equ     18      *表示する長さ
 8: WIDE            equ     4       *横に表示する数
 9: ********************************コマンドライン検査
10:         .text
11: OPTION_CHECK:
12:        `tst.b   (A2)            *なにか後についているか?
13:         beq     GET_CUR         *ついていない
14: ********************************ヘルプ表示
15: HELP:
16:         pea     HELP_MES
17:         DOS     _PRINT
18:         addq.l  #4,SP
19:         DOS     _EXIT
20: ********************************カーソル座標を求める
21: GET_CUR:
22:         IOCS    _OS_CUROF       *カーソルを消す
23:         moveq.l #-1,D1
24:         IOCS    _B_LOCATE
25:         move.w  D0,D2           *Y座標
26:         move.w  D2,CUR_Y
27: ********************************GET_CURDRIVE
28: GET_CURDRV:
29:         lea     CHDIR_BUF,A1
30:         DOS     _CURDRV
```

```
31:         addi.b  #'A',D0
32:         move.b  D0,(A1)+
33:         move.b  #':',(A1)+
34:         pea     CHDIR_BUF+3
35:         move.w  #0,-(SP)
36:         DOS     _CURDIR
37:         addq.l  #6,SP
38:         cmpi.b  #$00,CHDIR_BUF+3
39:         beq     GET_FILES       *ルートディレクトリだった
40:         move.b  #'¥',(A1)
41: ********************************ディレクトリ検索
42: GET_FILES:
43:         clr.w   D3
44:         lea     DIR_BUF,A2      *転送先アドレス(STRCPYで使う)
45:         move.w  #$10,-(SP)      *属性(ディレクトリ)
46:         pea     WILD
47:         pea     FILBUF
48:         DOS     _FILES          *スタック補正しない +10
49:         cmpi.l  #0,D0
50:         bmi     NOT_FOUND       *D0<0(エラー)
51:         bsr     STRCPY          *バッファに転送
52: START_NFILES:
53:         DOS     _NFILES
54:         cmpi.l  #0,D0
55:         bmi     NOT_NEXT        *D0<0
56:         cmpi.l  #MAX,D3
57:         beq     NOT_NEXT        *D3=MAX+1
58:         addq.w  #1,D3           *MAX=MAX+1
59:         bsr     STRCPY
60:         bra     START_NFILES
```


▶て，てーでんがこわい……。かみなりなんてキライだぁー(しくしく)。無停電電源装置がほしくなったなあ。
松本　勝正(19)富山県

```
 61: NOT_NEXT:
 62:         lea     10(SP),SP          *スタック補正(DOS_FILES +10)
 63:         move.w  D3,FOUND_MAX       *最大数格納
 64:         bra     PRINT              *PRINT
 65: **********************************STRCPY
 66: STRCPY:
 67:         move.l  #LEN,D4
 68:         lea     FILNAME,A1         *転送元アドレス(FILNAME)
 69: CPY_STR:                           *文字の間回る
 70:         subq.l  #1,D4
 71:         bmi     CPY_RETURN
 72:         cmpi.b  #$00,(A1)
 73:         beq     CPY_SPACE
 74:         move.b  (A1)+,(A2)+
 75:         bra     CPY_STR
 76: CPY_SPACE:                         *EOSを発見!!
 77:         addq.l  #1,A1
 78:         move.b  #$20,(A2)+         *$20=SPACE
 79:         subq.l  #1,D4
 80:         bmi     CPY_RETURN
 81:         bra     CPY_SPACE
 82: CPY_RETURN:
 83:         move.b  #$00,(A2)+         *$00=EOS
 84:         rts
 85: **********************************PRINT
 86: PRINT:
 87:         clr.w   D4
 88:         lea     DIR_BUF,A1         *表示する先頭番地
 89:         move.w  FOUND_MAX,D3       *最大数
 90:         move.b  #WIDE,D2           *横方向の表示数
 91: START_PRINT:
 92:         IOCS    _B_PRINT
 93:         subq.b  #1,D3              *MAX-1
 94:         bmi     END_PRINT          *MAX-1<0
 95:         subq.b  #1,D2
 96:         bmi     CUR_DOWN
 97:         bra     START_PRINT
 98: CUR_DOWN:
 99:         bsr     PRINT_CRLF
100:         move.b  #WIDE,D2           *横方向の表示数
101:         addq.w  #1,D4
102:         bra     START_PRINT
103: END_PRINT:
104:         bsr     PRINT_CRLF
105:         lea     HOWTO_MES,A1
106:         addq.w  #1,D4
107:         IOCS    _B_PRINT
108: CUR_HOSEI:
109:         moveq.l #-1,D1
110:         IOCS    _B_LOCATE
111:         move.w  D0,END_Y           *終了時のY座標
112:         move.w  CUR_Y,D5
113:         add.w   D5,D4
114:         subi.w  #30,D4
115:         bmi     GET_DIR
116:         sub.w   D4,CUR_Y
117: **********************************GET_DIR
118: GET_DIR:
119:         clr.w   D4
120:         clr.w   D5
121:         clr.w   D6
122:         move.w  #0,D1              *座標X
123:         move.w  CUR_Y,D2           *座標Y
124:         move.b  #1,D3              *属性(0=普通/1=反転)
125:         bsr     PRINT_GET_DIR
126: INPUT:                             *キー判別
127:         DOS     _INKEY             *キーコード読みだし
128:         move.l  d1,a6
129:         moveq.l #0,d1
130:         IOCS    _BITSNS
131:         move.l  a6,d1
132:         btst    #01,d0             *ESC
133:         bne     QUIT
134:
135:         move.l  d1,a6
136:         moveq.l #9,d1
137:         IOCS    _BITSNS
138:         move.l  a6,d1
139:         btst    #06,d0             *ENTER
140:         bne     CHDIR
141:         move.l  d1,a6
142:         moveq.l #3,d1
143:         IOCS    _BITSNS
144:         move.l  a6,d1
145:         btst    #05,d0             *CR
146:         bne     CHDIR
147:
148:         move.l  d1,a6
149:         moveq.l #7,d1
150:         IOCS    _BITSNS
151:         move.l  a6,d1
152:         btst    #03,d0             *LEFT
153:         bne     LEFT
154:         btst    #05,d0             *RIGHT
155:         bne     RIGHT
156:         btst    #04,d0             *UP
157:         bne     UP
158:         btst    #06,d0             *DOWN
159:         bne     DOWN
160:         bra     INPUT
161: UP:
162:         cmpi.w  #WIDE+1,D5
163:         bmi     INPUT              *D5<WIDE+1
164:         move.b  #0,D3              *属性(0=普通/1=反転)
165:         bsr     PRINT_GET_DIR
166:         subq.w  #WIDE+1,D5         *番号-WIDE+1
167:         subq.l  #1,D2              *Y座標-1
168:         bra     GOOD
169: DOWN:
170:         addq.w  #WIDE,D5
171:         cmp.w   FOUND_MAX,D5
172:         bcc     DOWN_BAT
173:         move.b  #0,D3              *属性(0=普通/1=反転)
174:         subq.w  #WIDE,D5
175:         bsr     PRINT_GET_DIR
176:         addq.w  #WIDE+1,D5         *番号+WIDE+1
177:         addq.l  #1,D2              *Y座標+1
178:         bra     GOOD
179: DOWN_BAT:
180:         subq.w  #WIDE,D5
181:         bra     INPUT
182: LEFT:
183:         tst.w   D5
184:         beq     INPUT              *D5=0
185:         move.b  #0,D3              *属性(0=普通/1=反転)
186:         bsr     PRINT_GET_DIR
187:         subq.w  #1,D5              *番号-1
188:         cmpi.w  #0,D4
189:         beq     RLF
190:         subi.w  #LEN,D1            *X座標-LEN
191:         subq.l  #1,D4
192:         bra     GOOD
193: RLF:
194:         move.w  #LEN*WIDE,D1
195:         subq.w  #1,D2              *Y座標-1
196:         moveq.l #WIDE,D4
197:         bra     GOOD
198: RIGHT:
199:         cmp.w   FOUND_MAX,D5       *D5-FOUND_MAX
200:         beq     INPUT              *D5-FOUND_MAX=0
201:         move.b  #0,D3              *属性(0=普通/1=反転)
202:         bsr     PRINT_GET_DIR
203:         addq.w  #1,D5              *番号+1
204:         cmpi.w  #WIDE,D4           *D4-WIDE
205:         beq     LF                 *D4-5=0
206:         add.w   #LEN,D1            *X座標+LEN
207:         addq.l  #1,D4
208:         bra     GOOD
209: LF:
210:         move.w  #0,D1
211:         addq.w  #1,D2              *Y座標+1
212:         moveq.l #0,D4
213: GOOD:
214:         move.b  #1,D3              *属性(0=普通/1=反転)
215:         bsr     PRINT_GET_DIR
216:         bra     INPUT
217: QUIT:
218:         bra     EXIT
219: **********************************PRINT_GET_DIR
220: PRINT_GET_DIR:
221:         IOCS    _B_LOCATE
222:         lea     DIR_BUF,A1
223:         move.w  D1,-(SP)
224:         move.w  D5,-(SP)
225: ADRESS:
226:         subq.w  #1,D5
227:         bmi     ADRESS_OK
228:         add.l   #LEN+1,A1          *A1=A1+LEN+1
229:         bra     ADRESS
230: ADRESS_OK:
231:         tst.b   D3
232:         beq     FUTUU
233:         move.w  #%0000_1010,D1     *黄色リバース
234:         IOCS    _B_COLOR
235:         bra     ZOKUSEI_OK
236: FUTUU:
237:         move.w  #%0000_0011,D1     *白
238:         IOCS    _B_COLOR
239: ZOKUSEI_OK:
240:         IOCS    _B_PRINT           *表示
241:         move.w  (SP)+,D5
242:         move.w  (SP)+,D1
243:         rts
244: **********************************CHDIR
245: CHDIR:
246:         sub.l   #LEN+1,A1
247:         lea     CHDIR_BUF,A2
248: FOUND_EOS:
249:         cmpi.b  #$00,(A2)+
250:         bne     FOUND_EOS
251:         subq.l  #1,A2
252:         move.b  #'¥',(A2)+
253: CHDIR_BUF_CPY:
254:         cmpi.b  #$20,(A1)
255:         beq     CHDIR_CPY_END
256:         move.b  (A1)+,(A2)+
257:         bra     CHDIR_BUF_CPY
258: CHDIR_CPY_END:
259:         pea     CHDIR_BUF
260:         DOS     _CHDIR
261:         addq.l  #4,SP
262:         bra     EXIT
263: **********************************PRINT_CRLF
264: PRINT_CRLF:
265:         pea     CRLF
266:         DOS     _PRINT
267:         addq.l  #4,SP
268:         rts
269: **********************************NOT_FOUND
270: NOT_FOUND:
271:         pea     NOT_FOUND_MES
272:         DOS     _PRINT
273:         addq.l  #4,SP
274:         move.w  CUR_Y,END_Y
275: **********************************EXIT
276: EXIT:
277:         move.w  #%0000_0011,D1     *属性白
278:         IOCS    _B_COLOR
279:         move.w  #0,D1
280:         move.w  END_Y,D2
281:         IOCS    _B_LOCATE
282:         bsr     PRINT_CRLF
```

▶ついにBJプリンタを買ってしまった。なに！ ドイツ語のフォントを持ってるのか!?
しかし、エディタがねーな。　栢 一夫(19)三重県

```
283:            IOCS    _OS_CURON             *カーソルを表示する
284:            DOS     _EXIT
285: ********************************DATAエリア
286:            .data
287:            .even
288: CRLF:
289:            .dc.b   $0D,$0A,0             *改行コード
290: WILD:
291:            .dc.b   '*.*',0               *ワイルドカード
292: NOT_FOUND_MES:                           *タイトル
293:            .dc.b   $1b,'[36mディレクトリは存在しません'
294:            .dc.b   $1b,'[33m',$0D,$0A,0
295: HELP_MES:                                *タイトル
296:            .dc.b   $1b,'[36mX68k XCD ver1.00'
297:            .dc.b   $1b,'[33m Copyright 1993 K.O',$0D,$0A,0
298: HOWTO_MES:                               *使用方
299:            .dc.b   $1b,'[35mカーソル:選択'
300:            .dc.b   $1b,'[33m / ',$1b,'[35mRETURN:決定'
301:            .dc.b   $1b,'[33m / ',$1b,'[35mESC:キャンセル',0
302:            .bss
303:            .even
304: CUR_Y:
305:            .ds.w   1
306: END_Y:
```

```
307:            .ds.w   1
308: FOUND_MAX:                               *見付けた数
309:            .ds.w   1
310: FILBUF:                                  *FILES/NFILES用
311:            .ds.b   1
312:            .ds.b   1
313:            .ds.w   1
314:            .ds.w   1
315:            .ds.w   1
316:            .ds.w   1
317:            .ds.b   8
318:            .ds.b   3
319:            .ds.b   1
320:            .ds.w   1
321:            .ds.w   1
322:            .ds.l   1
323: FILNAME:
324:            .ds.b   23
325: CHDIR_BUF:
326:            .ds.b   67
327: DIR_BUF:
328:            .ds.b   (LEN+1)*MAX           *最大検索数分のバッファ
329: ********************************
330:            .end
```

ぱーてぃハンズ(9)

前々回ではプレイヤー1のルーチンをきれいに書き直したんでしたね。今回はまず，同じようにプレイヤー2のルーチンを書きます。リストでいうと1560行〜です。なんのことはない，プレイヤー1のプログラムから，変数名をたとえば，dx1という変数であればdx2というように書き換えたものをつけ加えただけなんですけどね(前にプレイヤー2のルーチンをつけたときと同じ手口ですね)。

変数を工夫してこのプレイヤー1とプレイヤー2の入力ルーチンのように，同じようなルーチンはひとつにまとめるのが本当は美しいんでしょうけど，なにしろX-BASICは配列のアクセスが遅いんですよね。で，今回はこれで済ましてしまうのでした。

さて，これであっさり2人プレイのゲームはできたので，これから我がぱーてぃハンズもいよいよ大詰め，当たり判定の処理をつけます。

そして当たり判定なのだ！

編集室にもよくいろいろなソフトハウスさんからゲームのサンプルが送られてきますけど，ときどき「もうすぐ完成版ですよ」ってことで当たり判定のまだついていないバージョンが送られてくることがあります。

ゲームにとって，特にシューティングゲームや今回の格闘ゲームのようにキャラクターとキャラクターが当たることでゲームになるものでは，当たり判定は最後につけることが多いみたいです。当たり判定でゲームの良し悪しが決まってしまいますので，当然といえば当然なんですけどね。ちなみにロールプレイングゲームでは，戦闘場面が最後になることが多いみたいです。アドベンチャーは……よくわからないですけど，あえていうならシステムを最初に作ってあとは順番に場面を増やしていく……かなぁ。

うおりゃー，キックだ！

まず，当たり判定をいつやりましょうか。キャラクターが2人いて殴る蹴るするんですから，一番必要なのは殴ったり蹴ったりするときですよね。プレイヤー1の場合であれば1080行で，

　if(strig(1)=1) then 〜

と，ジョイスティックのボタンが押されたかどうかを見て，殴ったり蹴ったりっていうポーズのスプライトを表示しています。とりあえずここで当たり判定をつけましょう。

んーっと，殴ったか蹴ったかでちょっと処理を変えたいので，goto文で飛んでいく前に，

　atari(n)

と変数をつけて当たり判定ルーチンに飛ばします。呼び出す側はこれでおっけーです。

踏み込んで殴るのだ！

さてさて呼び出されるほうの当たりルーチン本体です。その前にキャラクターの表示についてちょっと。1080，1090行の

　if(strig(1)〜

でジョイスティックのトリガが押されたかどうかを見てるんでしたよね。このときにPutSprite()でキャラクターを表示させているのですが，この殴ったときや蹴ったときのスプライトのパターンって……。そう，スプライトのパターンいっぱいに描いているせいで，このPutSprite()でそれまで表示して殴ったパターンを表示すると，ずずっと後ろに下がったようになっちゃうんですよ。でも本当は殴るときとか蹴るときってのは力んでいるわけだから逆に一歩踏み出さなくちゃいけないんです。ってーことで，zxnっていう変数を作って，キャラクターの向きによってこのzxnにキャラクターの踏み込むドット数分の数値を入れて，トリガが押されたときはこのzxn変数をキャラクターの位置に足して，踏み込んだように見せることにしました。1080，1090行のあたりを見てください。うーむ，またひとつif文と変数が増えてしまいました。

さて，atari()関数の中身です。

当たり判定ですけども，どういうふうに当たりを判定してそれで何をするか考えましょう。

まず，トリガが押されているときには，自分は相手を殴ろうとして前に出ているんですよね。そのとき自分のキャラクターはプレイヤー1であればzx1ドット分前にいるんですよね。このときに敵のキャラクターがこの手の先にいて，高さが同じになっていればいいんですね。っていうことは，まず，自分のキャラの手の先はまず，自分の左端がdx1，で，キャラクターの大きさは32ドットありますから，右向きのときには座標を＋31していって，さらに踏み込んでますからプラス＋zxnしたところがキャラクターの右端の位置です。これが敵キャラのスプライトの左端の座標dx2より右，

　x1＋31＋zx1＞dx2

でもって敵のキャラの右端より左側にいなくてはならないので，

　x1＋31＋zx1＜dx2＋31

の両方の条件が成り立てばいいんですよ。ってことは，

　if((x1＋31＋zx＞dx2)and(x1＋31＋zx＜dx2＋31))

でみてやればいいんですね。

で，さらにキャラクターが逆の方向を向いている場合にはzx1は踏み込んだ分はちゃーんとマイナスになってますからそのままでいいでしょう。で，さっき基準をキャラクターの右端からにするために，＋31していたのを0にしてしまえばいいんですね。キャラクターが右向きか左向きかはd1という変数に0か1かで入っているんですよね。キャラが右向きのときは0，左向きのときは1です。

　d1＝0のとき31

　d1＝1のとき0

……逆に1のとき31で，0のとき0にするなら簡単ですよね。d1＊31でいいんですから。それじゃあ，d1が0のとき1になって1のとき0になるようにして，31をかければ出したい数になりますよね。

そう，これは前に使ったxorを使えばいいのです。0 xor 1=1，1 xor 1=0ですから，(d1 xor 1)＊31で求めたい数になるわけです。

これを使って

　x1＋(d1 xor 1)＊31 ＋zx1

で，求めたい手の先の座標になるのです。わか

ったかな？

殴られると痛いんです～

で，当たったあとの処理です。

当たったときには敵がダメージを受けなくてはいけません。まだスコアは全然作っていませんでしたね。ここでは，よくある格闘ゲームと同じようにヒットポイント制にしてしまいましょう。hp1，hp2という変数を作ってこれをヒットポイントにします。まず最初の処理で両方のキャラクターのヒットポイントを10にします。

で，この当たったときにパンチで1，キックで2ダメージを与えます。で，敵のヒットポイントが0以下になったときには，自分の勝ちにします。もうこの辺の処理は簡単ですよね。たんに引算してif文でhp2が0以下かどうか調べて，その処理をするだけですね。

ついでに敵が打たれたことをわかりやすくするために，敵位置をちょっとずらしましょう。そう，敵に殴られて後ろによろっといくわけです。敵の座標はx2ですから，ここではx2=x2＋zx1としてやれば簡単にできそうですね。

で，実際にできたのが10000行以降のatari1()，atari2()関数が当たり処理ルーチンです。キャラクタがスプライトいっぱいに描かれているわけではないので，実際の当たり判定は，ちょっと範囲を狭くしてあります。

できた！

さーて，やっとひと通りゲームの形になったところで遊び方の説明でも書いておきましょうかね。

FIGHT.BAS for X68000/030

（要X-BASIC，ジョイスティック，XVI以上推奨）

このゲームは対戦専用の格闘ゲームです。ゲームをするには2本のジョイスティックが必要です。

2人のキャラクターが画面に表示されます。左側がプレイヤー1，右側がプレイヤー2です。ジョイスティックの左右でキャラクターが左右に動き，上に入れるとジャンプします。Aボタンでパンチ，Bボタンでキックしますのでこれで敵を倒してください。プレイヤー1，2どちらかのヒットポイントがなくなったところでゲームオーバーです。

BASICインタプリタ上で遊ぶには，XVI以上のほうがいいですね。コンパイルする場合には適当にウエイトを入れておいてくださいね。

はぁ，これでやっとひと通りゲームが形になりましたっ！　いや～，よかったよかった。

結局XVI以上でないと遊ぶにはちょっと苦しいプログラムになってしまいましたけれども，勘弁してください。

えーっと，このプログラムは本当に本当に，まだひと通りとりあえずやった，というレベルにしか来てません。たとえば，「スト2」などだとある，キャラとキャラが押し合いしてずりずり動かせるとか，効果音とかですね。

しかし，キャラの押し合いというのは基本的にキャラの当たり判定ですし，効果音も当たり判定のところで音を出させればいいだけですしね(ただし，押し合いの当たり判定は始終チェックしなきゃならんので，結構重い作業ですね。コンパイル必須になってしまいますね)。この辺は，ぜひとも皆さんでつけてみてくださいね。

図1　殴るときはz×1ドット右へ

図2　攻撃が当たっているときの座標

リスト

```
 1000 screen 0,0,1,0
 1010 int x1,y1,h1,dx1,k1,l1,dh1,d1,zx1
 1015 int x2,y2,h2,dx2,k2,l2,dh2,d2,zx2
 1020 x1=48:y1=128:dh1=-4:l1=0:d1=0:hp1=10
 1025 x2=208:y2=128:dh2=-4:l2=0:d2=1:hp2=10
 1030 sp_disp(1)
 1040 sp_on(0,18):p=0
 1045 for i=0 to 127:sp_move(i,-15,-15):next
 1050 PutSprite(0,0,x1,y1,0):PutSprite(0,1,x2,y2,1)
 1060 locate 3,0:print"1P HP=";:print hp1
 1061 locate 20,0:print"2P HP=";:print hp2
 1062 if (k1<>0) then { k1=k1-1
 1065 }else{
 1070   if(h1=0 and l1=0) then{
 1075     if(d1=0) then zx1=4 else zx1=-4
 1080     if(strig(1) = 1) then PutSprite(2,0,x1+zx1,y1,d1):k1=2
:l1=1:atari1(1):goto 1230
 1090     if(strig(1) = 2) then PutSprite(4,0,x1+zx1,y1,d1):k1=4
:l1=1:atari1(2):goto 1230
 1100     switch (stick(1))
 1105       case 5: break
 1110       case 4: walk(0,x1,y1):x1=x1-4:d1=1:break
 1120       case 6: walk(0,x1,y1):x1=x1+4:d1=0:break
 1130       case 7: h1=-8:dh1=-8:dx1=-4:d1=1:break
 1150       case 9: h1=-8:dh1=-8:dx1=+4:d1=0:break
 1160       case 8: h1=-8:dh1=-8:dx1=0:break
 1170     endswitch
 1180   }else{
 1190     if(strig(1) = 1 and h1=0 and l1=1) then PutSprite(0,0,
x1,y1,d1):goto 1230
 1200         if(strig(1) = 2 and h1=0 and l1=1) then PutSprite(
0,0,x1,y1,d1):goto 1230
 1210     }
 1220  if(h1=0) then PutSprite(0,0,x1,y1,d1):l1=0
 1225 }
 1230 if(x1<16)  then { dx1=dx1*-1:x1=x1+dx1:x1=16 }
 1240 if(x1>224) then { dx1=dx1*-1:x1=x1+dx1:x1=224 }
 1250 if(h1<>0) then h1=h1+dh1:x1=x1+2*dx1:PutSprite(6,0,x1,y1+h
1,d1):if(h1<-64) then dh1=8
 1270 /*goto 1060*/
 1560 if (k2<>0) then { k2=k2-1
 1565 }else{
 1570   if(h2=0 and l2=0) then{
 1575     if(d2=0) then zx2=4 else zx2=-4
 1580     if(strig(2) = 1) then PutSprite(2,1,x2+zx2,y2,d2):k2=2
:l2=1:atari2(1):goto 1730
 1590     if(strig(2) = 2) then PutSprite(4,1,x2+zx2,y2,d2):k2=4
:l2=1:atari2(2):goto 1730
 1600     switch (stick(2))
 1605       case 5: break
 1610       case 4: walk(1,x2,y2):x2=x2-4:d2=1:break
 1620       case 6: walk(1,x2,y2):x2=x2+4:d2=0:break
 1630       case 7: h2=-8:dh2=-8:dx2=-4:d2=1:break
 1650       case 9: h2=-8:dh2=-8:dx2=+4:d2=0:break
 1660       case 8: h2=-8:dh2=-8:dx2=0:break
 1670     endswitch
 1680   }else{
 1690     if(strig(2) = 1 and h2=0 and l2=1) then PutSprite(0,1,
x2,y2,d2):goto 1730
 1700         if(strig(2) = 2 and h2=0 and l2=1) then PutSprite(
0,1,x2,y2,d2):goto 1730
 1710     }
 1720  if(h2=0) then PutSprite(0,1,x2,y2,d2):l2=0
 1725 }
 1730 if(x2<16)  then { dx2=dx2*-1:x2=x2+dx2:x2=16 }
 1740 if(x2>224) then { dx2=dx2*-1:x2=x2+dx2:x2=224 }
 1750 if(h2<>0) then h2=h2+dh2:x2=x2+2*dx2:PutSprite(6,1,x2,y2+h
2,d2):if(h2<-64) then dh2=8
 1770 goto 1060
 9510 func walk(player,x, y)
 9520 int ix,iy
 9530   for  ix=0 to 1
 9540     iy=2
 9550     sp_set(ix+iy*2+6*player,x+ix*16,y+iy*16,8 +ix+iy*8 + 2
56)
 9560   next
 9570 endfunc
 9580 func PutSprite(pnum,player, x, y, d)
 9590 int ix,iy
 9600   for  ix=0 to 1
 9610     for iy=0 to 2
 9620 sp_set(ix+iy*2+6*player,x+(ix xor d)*16,y+iy*16,pnum+ix+iy
*8 + 256 +16384*d)
 9630     next
 9640   next
 9650 endfunc
10000 func atari1(hit)
10020   if((x1+(d1 xor 1)*31+zx1>x2 + 6) and (x1+(d1 xor 1)*31+z
x1<x2+25) and (y1=y2)) then{
10030      hp2 = hp2 - hit:x2=x2+zx1
10040   }
10050   if(hp2<=0) then locate 10,3 :print"Player1 Win!!!":locat
e 0,12:stop
10999 endfunc
20000 func atari2(hit)
20020   if((x2+(d2 xor 1)*31+zx2>x1 + 6) and (x2+(d2 xor 1)*31+z
x2<x1+25) and (y1=y2)) then{
20030      hp1 = hp1 - hit:x1=x1+zx2
20040   }
20050   if(hp1<=0) then locate 10,3 :print"Player2 Win!!!":locat
e 0,12:stop
20999 endfunc
```

バックナンバー案内

ここには1992年10月号から1993年９月号までをご紹介しました。現在1992年６，７，９，12，1993年４～９月号の在庫がございます。バックナンバーはお近くの書店にご注文ください。定期購読の申し込み方法は160ページを参照してください。

1992

10月号（品切れ）
特集　DTMへの招待

連載　DōGA CGアニメーション講座/大人のためのX68000
響子 in CGわ～るど/吾輩はX68000である/ショートプロ
マシン語プログラミング/ハード工作/ANOTHER CG WORLD

●試用レポート　X68000用CD-ROMドライブ

LIVE in '92　美少女戦士セーラームーン/笑顔を探して 他

THE SOFTOUCH　ポピュラスII/リーディングカンパニー/ネクタリス/サークII

全機種共通システム　実践Small-C講座(6)SLENDER HUL

11月号（品切れ）
特集　ゲームマネージメント

連載　DōGA CGアニメーション講座/大人のためのX68000
響子 in CGわ～るど/ショートプロ/よいこのSX-WINDOW
ハード工作/ANOTHER CG WORLD/Computer Music入門

●新製品紹介　CHART PRO-68K

LIVE in '92　ストリートファイターII/スーパーマリオ 他

THE SOFTOUCH　キャッスルズ/シュートレンジ/ポピュラスII/サンダーレスキュー

全機種共通システム　実践Small-C講座(7)EDIT

12月号
Oh!X５周年特別企画　ショートプロ大集合

連載　DōGA CGアニメーション講座/マシン語プログラミング
響子 in CGわ～るど/ショートプロ/よいこのSX-WINDOW
大人のためのX68000/ハード工作/Computer Music入門

●エレクトロニクスショウ'92

LIVE in '92　LAST CHRISTMAS/闇の血族/ユーフォリー

THE SOFTOUCH　デスブレイド/ムーンクレスタ＆テラクレスタ/ふしぎの海のナディア/ロードス島戦記II 他

全機種共通システム　実践Small-C講座(8)MAKE

1993

1月号（品切れ）
特集　D.I.Y.ハードウェア

連載　DōGA CGアニメーション講座/マシン語プログラミング
響子 in CGわ～るど/ショートプロ/よいこのSX-WINDOW
大人のためのX68000/ハード工作/Computer Music入門

●新製品紹介　サンダーワード/SX広辞苑

LIVE in '93　ムーンライト伝説/チャコの海岸物語

THE SOFTOUCH　オーバーテイク/ストライダー飛竜/エアーマネジメント/パイプドリーム 他

全機種共通システム　実践Small-C講座(9)EDC-Tの拡張

2月号（品切れ）
特集　画像創造のために

連載　DōGA CGアニメーション講座/マシン語プログラミング
響子 in CGわ～るど/ショートプロ/よいこのSX-WINDOW
ハード工作/吾輩はX68000である/Computer Music入門

●新製品紹介　Communication SX-68K

LIVE in '93　FIRE CRACKER/サンバDEグワッシャ!

THE SOFTOUCH　極/ドラゴンスレイヤー英雄伝説/機甲装神ヴァルカイザー/キングス・ダンジョン

全機種共通システム　BLACK JACK

3月号（品切れ）
特集　X-BASICを学ぶ

連載　DōGA CGアニメーション講座/マシン語プログラミング
響子 in CGわ～るど/ANOTHER CG WORLD/ハード工作
ショートプロ/Computer Music入門/Z80's Bar

●緊急速報　32ビットマシンX68030

●新製品紹介　音源モジュールSC-33/GS音源搭載JW-50

LIVE in '93　ストリートファイターII/晴れたらいいね 他

THE SOFTOUCH　究極タイガー/チェルノブ/シムアント 他

全機種共通システム　シューティングゲームコアシステム作成法(1)

4月号
特集　X68第7世代へ

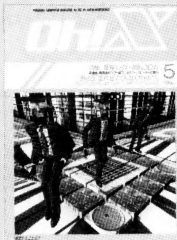
連載　DōGA CGアニメーション講座/マシン語プログラミング
響子 in CGわ～るど/ショートプロ/よいこのSX-WINDOW
ハード工作/吾輩はX68000である/Computer Music入門

●決定!　1992年GAME OF THE YEAR

●名作ゲーム再遊記

LIVE in '93　FIGHTMAN/ミンキーモモより 愛しのマーシカ

THE SOFTOUCH　スターフォース/元朝秘史 他

全機種共通システム　シューティングゲームコアシステム作成法(2)

5月号
特集　襲撃! SX-WINDOW
第８回　言わせてくれなくちゃだワ

連載　DōGA CGアニメーション講座/ANOTHER CG WORLD
響子 in CGわ～るど/ショートプロ/大人のためのX68000
ハード工作/吾輩はX68000である/Computer Music入門

●X68030へのソフトウェア対応について

LIVE in '93　MAGICAL SOUND SHOWER/もう笑うしかない 他

THE SOFTOUCH　エトワールプリンセス/メガロマニア 他

全機種共通システム　シューティングゲームコアシステム作成法(3)

6月号
創刊11周年特別企画　確率遊技シミュレーション

連載　DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ～るど/ショートプロ/大人のためのX68000
ハード工作/吾輩はX68000である/Computer Music入門

●新製品紹介　SC-55mkII

LIVE in '93　ストリートファイターIIより 春麗のテーマ/BAY YARD/LOVE&CHAIN

THE SOFTOUCH　餓狼伝説/信長の野望・覇王伝 他

全機種共通システム　REVERSI

7月号
特集　席巻するローテク文明

連載　DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ～るど/ショートプロ/マシン語プログラミング
ハード工作/吾輩はX68000である/Computer Music入門

●新製品紹介　ドローイングバット33070&MATIER

LIVE in '93　Midnight Circle/今日の日はさようなら/赤い靴

THE SOFTOUCH　悪魔城ドラキュラ/リブルラブル/大航海時代II/銀河英雄伝説III/幻影都市/ヴェルスナーグ戦乱

全機種共通システム　MSX用S-OS "SWORD"

8月号
特集　Ｃ言語実践的入門

連載　DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ～るど/Computer Music入門/大人のためのX68000
吾輩はX68000である/ショートプロ/ANOTHER CG WORLD

●特別企画　夏真っ盛り，アマチュアリズムのX68000

LIVE in '93　SPLASH WAVE

THE SOFTOUCH　悪魔城ドラキュラ/リブルラブル/餓狼伝説/ロボットコンストラクションR.C./Winning Post

全機種共通システム　MACINTO-C再掲載

9月号
特集　光学式磁気円盤MO

連載　DōGA CGアニメーション講座/こちらシステムX探偵事務所
響子 in CGわ～るど/ショートプロ/大人のためのX68000
ハード工作/Computer Music入門/ANOTHER CG WORLD

●新製品紹介　OS-9/X68030

LIVE in '93　ファイナルファンタジーVのテーマ/銀河鉄道999/アルスラーン戦記IIより 汗血公路/ちょうちょ

THE SOFTOUCH　悪魔城ドラキュラ/コットン/ダーク・オデッセイ 他

全機種共通システム　７並べ/SLANG再々掲載

AFTER REVIEW

10年の歳月を超え，あの「リブルラブル」がX68000に帰ってきました。当時生まれたばかりの赤ん坊ももう小学生，バシシの快感を覚えるには十分な年齢です。どうでもいいけど赤がリブルで青がラブルだよん。

リブルラブル

▶右脳と左脳のバランスがよくなりそう。
高橋　毅(22)埼玉県

▶テストの結果をズタズタに引き裂く，悪魔である。　江城　憲之(17)大阪府

▶生まれて初めてである。私にケガをさせたゲームは。右親指は十字キーを嫌い，血マメを……。　中矢　史朗(22)愛媛県

▶ほかのゲームにないコンセプトがいい。パッドもよろしい！
利川　章庸(25)神奈川県

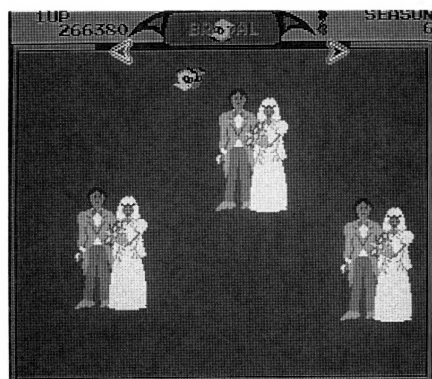
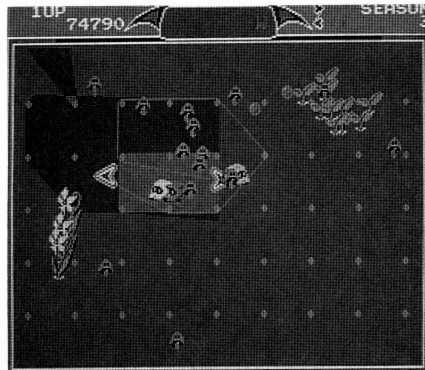
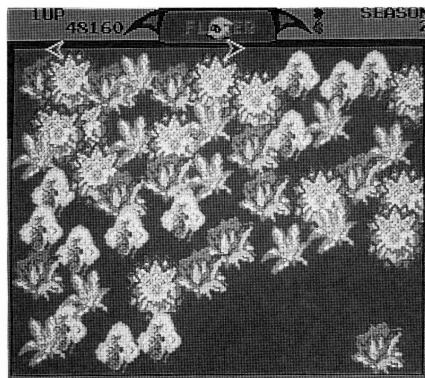
▶ひとりでやると気分がヘンになるゾ！(意味不明)　八木澤　良二(19)栃木県

▶私がまだまだ毛も生えてないガキの頃のあのバシシがまたできるなんて！　バシシ，バシシ，バシシ，本当にノスタルジー。
飯田　康博(25)愛知県

▶バシシはハシシ以上の快感だ。
林　寛(19)愛知県

▶熱中する前にゲーセンから消えたゲーム。あのときのくやしさが……幸せです。
諸岡　明(23)神奈川県

▶操作法に慣れていくにつれて深みにはまってしまう　高木　奈津彦(20)長野県

▶初体験。きゃー。　国政　寛(22)京都府

▶X68000を持っていることが自慢できる。
中島　太郎(21)神奈川県

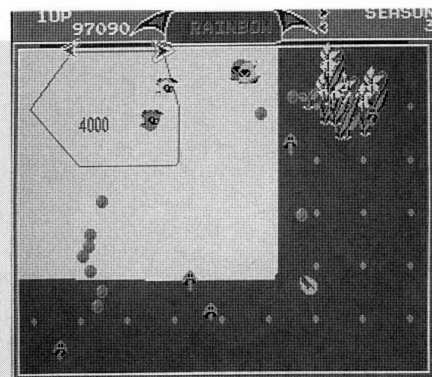
▶５年ぐらい昔，１時間も歩いてやりに行った。バス代がなかったんだよー。うれしーぜー。　広野　徹(19)東京都

▶う～，ムズイ。なかなか奇跡を起こせない。左右がこんがらかって，自らホブリンに突っ込む私であった……。
遠藤　正彦(22)岐阜県

▶最近，家内が左のレバー，私が右のレバーを操作してゲームをしています。こんな変則的なやり方も面白いですよ。
北原　純(30)愛知県

▶ナムコの見た夢によりかかって昼寝する。
中島　民哉(22)埼玉県

▶６月25日に買いました。１ゲーム目１面の宝箱の中は10万点クラウンでした。今年はついてるぜ。このゲームは大学受験の直前(1989年)に，近くのゲーセンに20円で置いてあって，はまったやつです。毎日，10ゲームだけと決めて，10回のうちに必ず10万点クラウンを出して「これで大学受かるぜ」と縁起をかついでいたゲームです。今年は大学院受験，10万点クラウンが出ますように。　松嶋　竜(23)東京都

▶よかった，同じだ！
古橋　康宏(18)栃木県

▶もうゲーセンにはありませんからねぇ。「これをやらずに死んだらまようぞ！」誰がいったか……名言だと思います。
佐藤　崇(26)栃木県

▶手も頭もパニック！越智　亮(20)大阪府

▶「移植が決定」の声を聞いてビックリ！そして出来栄えを見てさらにビックリ！ああ，このゲームがついに我が家でできるとは……。私は嬉しさのあまり「目蒲線の女」を歌いながら緑色の美しい画面に酔いしれるのでした。　横山　純一(18)東京都

▶私は「リブルラブルテクニカルマニュアル」も「バシシマーカー」も「バシシブック」(豆本)も持っている。そして幸運なことにX68000ユーザーである。こ，これで買わねば自分にウソをつくことになるっ！ 10年の時を超えていま，あの記憶がよみがえる……うるうる……。木島　智(23)東京都

▶懐かしいのに新鮮。右脳の鍛錬にもなる!?
北浜　慶(25)香川県

▶専用パッドがついてくる。オマケに弱い私。　藤村　良太(22)神奈川県

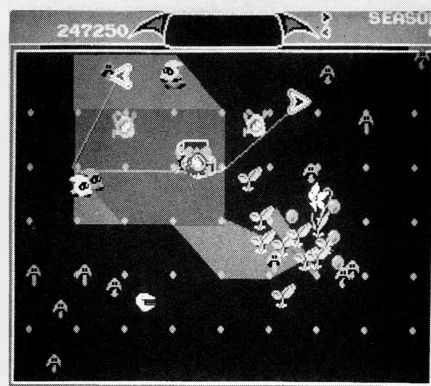
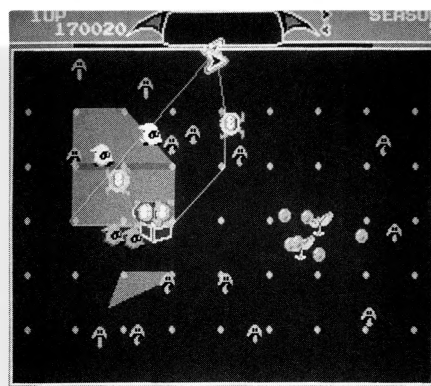
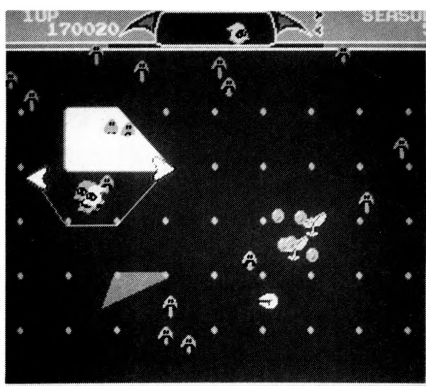
▶98.7%完全移植。NAME入力でアルファベットを動かし続けるとタイトルに戻らないところまで同じなんだぞ，すごい！
神谷　正樹(19)愛知県

▶両親指が痛い嬉しさ。この喜びは他機種のユーザーにはわかるまい。
小林　稔(20)東京都

▶独特な操作系。練り込まれたシステム。不器用な私は奇跡3回がいいところ。
芝田　稔(26)兵庫県

▶宝物が出てくると楽しくなる。
池田　雅敏(23)北海道

▶やられると必ずHi-SCOREが出るところまで移植されていた。
斉藤　学(19)神奈川県

▶「伝説のゲーム」にされてしまっていたが，初めて家庭でプレイできるようになり，

涙を流した人は多いことでしょう。これでまたX68000が売れているそうですよ。
川島　清史(23)神奈川県

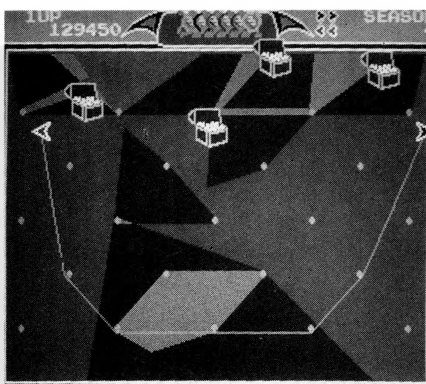
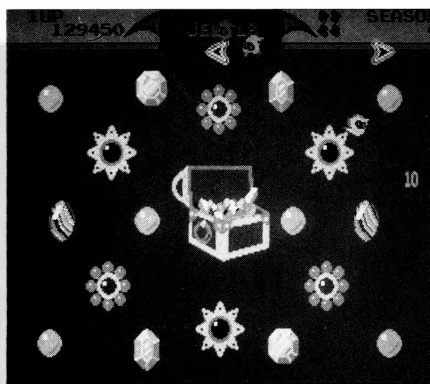
▶パッド目当てで買ったら，思いきりハマってしまった。両手の親指が痛い。
佐藤　充浩(22)福岡県

▶燃え尽きることのないロウソクとでも呼びたくなる，不思議な名作。それが「リブルラブル」であろう。待望のX68000版の登場で，幻だったこのゲームに触れ，アイデアがゲームを生み出していた時代の素晴らしさに，ゲームのなんたるかを考え直した人も，少なくないはずだ。

移植に対して厳しいことをいうと，ホブリンのプレイヤーのラインに対する反応や，

植物の生長や妨害キャラの頻度など，バランスに関わる違いなども挙げることができる。しかし，そういったことは些細なものと感じられるほど，ゲームシステム自体の魅力を無視できないところに，このゲームの真髄があるのだろう。

どこかで見たような，キャラクターを代えてバランスをいじっただけの作業で満足している素人は，これを遊んでゲームのなんたるかを，真面目に考えてもらいたい。冗談じゃなくて本気だ，うんうん。　(八)

▲上田考一　福岡県

クイズ大会

Murata Toshiyuki 村田 敏幸

7月号のクイズの解答はわかりましたか。この連載でもちょっとひと息ということで，今回は「68000クイズ大会」となりました。あなたの「68000」度はどれくらいか，日頃のウデを試してみてください。では，まず前回の解答から始めましょう。

7月号のコラムで68000についての軽いクイズを出してみたわけだが，今月はその解答編を拡大して，クイズ大会に走る。予告してあった“圧縮関係ネタ”は，原稿を書き始めてからすでに2カ月近く経っているにもかかわらずノリが悪くて文章にならないことから（正直でしょ）先送りとなった。

問題は前回の1問と合わせて大小6問。とくに点数配分を決めたりはしていないが，適当に自己採点して，自分がどのくらい68000に浸かっているか確認してみてほしい。なお，今回の問題では68000のみを考慮し，68010以降の上位プロセッサのことは考えない。また，命令の実行時間を示すときには，メモリアクセスにウェイトが入らないことを前提にする。

では，前回の問題とその解答から始めよう。

問1

以下の各定数値をd0.lに代入する最速の命令列を挙げなさい。速度が同じならそのうち最も総命令長が短いもの，速度も命令長も同じならそのうち最も命令数が少ないものを正解とする。

1) 255 $(000000FF_H)$
2) 65536 (00010000_H)
3) 32768 (00008000_H)
4) 2147483647 $(7FFFFFFF_H)$
5) 360 (00000168_H)
6) 256 (00000100_H)

【難易度低〜高】

32ビット即値の転送を速度優先で，サイズも考慮しつつ最適化する問題。ロングワード数値をそのままではなく，必要に応じてワード単位/バイト単位/ビット単位に分解して考えるのがポイントだ。となりに併記した16進表現が最大のヒントだろう。

だいたい難易度順に並べてあり，1)2)は基本のうち。即答を期待したい。3)は直接的な方法にこだわりすぎないこと。4)はちょっとひねっているが，選択肢が少ないので，順を追って考えれば答えに辿りつくと思う。5)は4)の変形。最後の6)は別に難しくはないものの，得られた答えが正しいことを確信するのに勇気が要るかもしれない。

問1解答

地道に筋道を立てて選択肢を狭めていくというアプローチをとろう。

まず，どんな値だろうと，12クロック/3ワード命令の，

move.l #data,d0

を使えばd0に転送できるわけだから，求める命令列は最悪でも12クロック以下となる。ここで，この12クロックという実行時間は3ワード命令としては最も速いので，2ワード以下の命令列でなければこれより速くはならない。また，データレジスタの全32ビットを高速に更新するにはmoveqの使用が必須だ。こうやって考えていくと，特定の32ビット即値をmove.lよりも速くd0に転送する方法があるとするなら，

moveq単独

という自明の（しかし，この問題には不適当な）ケースを除けば，

moveq＋1ワードの命令

という形しかありえないことがわかる。moveqの実行に必要な4クロックを差し引くと，この1ワード命令の実行時間は8クロック以下[1]。これだけの時間でできることといったら，アドレッシングモードをレジスタ直接形式とクイックイミディエイト形式に限ったデータ転送/算術演算/論理演算，および，ごく単純なビット操作ぐらいだ。加えて，レジスタをもう1本初期化するゆとりはないから，d0以外のレジスタを使うことも考えなくてよい。ここまで絞ったら，あとは個別対応でどうにかなる。

1) 255

解は4通り。できれば全部見つけてほしいところだが，実用上はひとつわかれば十分だろう。

$255=000000FF_H$をバイト単位に分解してみると，上位3バイトが00_H，最下位バイトがFF_Hとなる。上位バイトはmoveqでクリアできるから，とにかく最下位バイトをどうにかすればよい。そこで，このFF_Hをいろいろな視点から眺めてみる。

2の補数表現のつもりで見れば，FF_Hは-01_Hと等しい。ここから，

1) 8クロックちょうどのときには，move.lの実行時間と同じになるが，1ワード短いことから“最速のもののうち最短”という条件を満たしていることに注意。

```
        moveq.l #1,d0        (4/1)
        neg.b   d0           (4/1)
                          (計8/2)
```
および，
```
        moveq.l #0,d0        (4/1)
        subq.b  #1,d0        (4/1)
                          (計8/2)
```
の2つの解が得られる（カッコ内は実行に要するクロックサイクル数とワード単位の命令長)。前者は01_Hの2の補数をとる（＝符号反転する）ことで，後者は$-01_H=00_H-01_H$と考えて0から1を引くことで，FF_Hを作り出している。

また，FF_Hは00_Hの全ビットを反転したもの，言い換えると，00_HとFF_Hは互いに1の補数の関係にある。ここから，3つ目の解，
```
        moveq.l #0,d0        (4/1)
        not.b   d0           (4/1)
                          (計8/2)
```
が得られる。この2命令の組み合わせはなかなか重宝で，最初のmoveqで代入する値をいろいろ変えると，−256〜−129，＋128〜＋255の範囲の任意の値を作り出すことができる。moveq単独の場合と合わせれば，−256〜＋255の範囲が網羅できるわけだ。同様に，not.bの代わりにnot.wを使うと−65536〜−65409，＋65408〜＋65535の値が得られる。negを使っても似たようなことはできるが，notのほうがわずかに応用範囲が広いことを指摘しておこう。

残るひとつの解は，2進数で，

$$
\begin{array}{rl}
 01111111_B & (=7F_H) \\
+\ 10000000_B & (=80_H) \\
\hline
 11111111_B & (=FF_H)
\end{array}
$$

と考えると，
```
        moveq.l #127,d0      (4/1)
        tas.b   d0           (4/1)
                          (計8/2)
```
だ。tas命令は指定のバイトオペランドを0と比較して結果をccrに反映すると同時に第7ビットを立てるという動作をする。ここでは，後者の作用のみを利用し，80_Hを足す（正確には論理和をとる）のと同じ効果を得ている。

2) 65536

$65536=00010000_H$をワード単位に分解してみると，上位ワードは0001_H，下位ワードは0000_H。この上位下位を交換してできる00000001_Hならmoveq一発で転送できることに目をつけると，
```
        moveq.l #1,d0        (4/1)
        swap.w  d0           (4/1)
                          (計8/2)
```
という答えが得られる。最初のmoveqで仮に上位ワードを0000_H，下位ワードを0001_Hにしておいてから，swapでひっくり返すわけだ。同様の方法で，127倍までの65536の倍数，および，−128倍までの65536の倍数−1が作れる。

3) 32768

$32768=00008000_H$は第15ビットだけが1になる数だから，moveqで全ビットを0にしてから1ワード命令でこのビットを立てるという方向で考えてみよう。この用途に使えそうな命令はaddi，ori，bsetといったあたりだ。しかし，どの命令を使っても，
```
        addi.w #$8000,d0
        ori.w  #$8000,d0
        bset.l #15,d0
```
のようにソースオペランドにイミディエイト形式を適用せざるをえないため，1ワードには収まらない。そこでちょっと発想を変えて，最初のmoveqで全ビットを0にするのではなく，どこか1ビットだけ1にしておき，この1をシフトして第15ビットに持っていくことを考える。68000のシフト系命令は決して速い命令ではないが，ワード（またはバイト）サイズでデータレジスタを1ビットだけずらす場合の実行時間は8クロックサイクルであり，ぎりぎり条件を満たしている。ここで，第15ビットがワードデータの最上位ビットであることに気づけば，
```
        moveq.l #1,d0        (4/1)
        ror.w   #1,d0        (8/1)
                          (計12/2)
```
という答えが得られる。ローテート命令では最下位ビットと最上位ビットが繋がっているものとして扱われるので，第0ビットを立ててから右に1ビット回転してやるわけだ。2命令目は，
```
        ror.w   d0,d0        (8/1)
```
でもよい。

4) 2147483647

$2147483647=7FFFFFFF_H$は，第31ビットのみが0で残りのビットが1になる数だ。0と1が逆になってはいるものの，3)の32768と似たパターンといえる。とはいっても，3)の方法をそのまま応用した，
```
        moveq.l #-2,d0       (4/1)
        ror.l   #1,d0        (10/1)
                          (計14/2)
```
では，全32ビットをローテートする必要があるために，さっきよりも2クロック余計にかかる。ここは，3)では早々に捨てたストレートな考え方，つまり，
```
        moveq.l #-1,d0
```
により全32ビットを1にしてから，第31ビットだけを0にするという線に立ち戻るのが正しい。

利用できそうな命令はandiとbclr。
```
        andi.l  #$7fffffff,d0
```
は問題外なので，
```
        bclr.l  #31,d0
```
に注目する。実は，この命令は，
```
        bclr.l  d0,d0
```
に置き換えが可能だ。bclr.lのソースオペランドは下位5ビットのみが有効だということを思い出そう。事前にd0に代入した$-1=FFFFFFFF_H$の下位5ビットを取り出せば$1F_H=31$，ここに31が隠れているのだった。で，ここまで苦労して導いた，

```
        moveq.l #-1,d0     (4/1)
        bclr.l  d0,d0      (10/1)
                        (計14/2)
```
だが，まだちょっと遅い。最後にもうひとひねりした，
```
        moveq.l #-1,d0     (4/1)
        bchg.l  d0,d0      (8/1)
                        (計12/2)
```
が正解だ。68000ではbclrよりもbset/bchgのほうが常に2クロック分速いので，クリアをする代わりに，1→0と反転するのだった。

5) 360

ヒントのところでも書いたように，4)の考え方がそのまま利用できる。正解は，
```
        moveq.l #$68,d0    (4/1)
        bset.l  d0,d0      (8/1)
                        (計12/2)
```
だ。$360=168_H=68_H+100_H$と考え，68_Hを32ビットで代入してから第8ビットを立てる(68_Hの下位5ビットを取り出せば08_H)。bset の代わりに bchg を使ってもよい。

なお，68000のマニュアルでは，
```
        bset.l  d0,d0
```
の実行には最大で8クロックサイクル必要と記されているのでここでもそれに従ったが，実際には，操作対象ビットが0～15のときは2クロック短い6クロックサイクルで処理が完了する（実測値）。

6) 256

2ワードの命令列でd0.lを256にする方法はいくつか考えられる。しかし，そのうち最も速い
```
        moveq.l #64,d0     (4/1)
        lsl.w   #2,d0      (10/1)
                        (計14/2)
```
もmove.lにはかなわない。最速ということであれば，素直にmove.lで転送する
```
        move.l  #256,d0    (12/3)
                        (計12/3)
```
が正解だ（露骨な引っ掛け）。

ところで，設問では代入先をd0に限定しているが，現実のプログラムで，とにかく256を32ビット値としてレジスタに保持しておきたいだけであれば，データレジスタにはこだわらずに，アドレスレジスタを使うのが得策だ。つまり，アドレスレジスタに対してワード操作をすると自動的に32ビットに符合拡張されることを利用して，
```
        lea.l     256.w,a0   (8/2)
```
あるいは
```
        movea.w   #256,a0    (8/2)
```
とする。

問2

ccrのX，N，Z，V，Cの各ビットそれぞれについて，該当ビットを1にする最速の命令列をひとつずつ挙げなさい。また，該当ビットを0にする最速の命令列をひとつずつ挙げなさい。ccrのほかのビットは変化させても構わない。汎用レジスタも保存する必要はないが，変化させずにすむならそのほうがよい。　【難易度低】

サブルーチンの終了状態を返すのにccrの特定のビットを利用することがよくあるが，そんなときに有効な常套手段を求める問題。10パターンそれぞれについて個別に考える前に，常にccrが同じ変化を示す速い命令をいくつか考えてみるのが近道だろう。

問2解答

68000では4クロック命令が最速であり，moveq，および，オペレーションサイズがワードまでのデータレジスタ間算術/論理演算命令，あとはデータレジスタに対するaddq, subqといったあたりがこれに該当する。で，ccrの任意のビットのセット/リセット操作は，4クロック命令1つかその組み合わせで実現できる。解はおよそ無数にあるので，ここでは代表的なものを挙げよう。

なかでも，
```
        cmp.w d0,d0
```
は，ccrの変化が固定的で，しかも，レジスタの値を変えないので，なかなか重宝だ。同じ値同士を比較するわけだから，常にN＝0，Z＝1，V＝0，C＝0となる（Xビットはcmpの仕様上，変化しない）。この1命令で，いま求める10パターンのうち，4つがかたづく。

つぎに有用なのは，やはりmoveqだろう。
```
        moveq.l #1,d0
```
のように，とにかく非0の値を転送すればZ＝0になるし，
```
        moveq.l #-1,d0
```
など，負の値を転送すればN＝1になる。このmoveqにaddq/subqを組み合わせれば，
```
        moveq.l #0,d0
        subq.b  #1,d0
```
によりX＝1，C＝1，
```
        moveq.l #127,d0
        addq.b  #1,d0
```
によりV＝1が，
```
        moveq.l #0,d0
        addq.b  #1,d0
```
によりX＝0が実現できる。これで10パターンだ。

問3

nop (No OPeration) はpcを1ワード進める以外にはなんの副作用も持たない（＝ccrやレジスタ/メモリ内容を変化させない）4クロック/1ワードの命令だ。同様に4クロック/1ワードの何もしない命令を挙げなさい。　【難易度低】

実用上の意味はあまりない問題。68000の命令のうちccrを変化させない命令をすぐに思いつくかどうかにかかっている。

問3解答

68000では，アドレスレジスタに対する操作時にはccrが変化しない（アドレス比較命令であるcmpaを除く)。そのうちレジスタ内容を変化させないものを探せば，

movea.l a0,a0

と，

lea.l (a0),a0

が見つかる。念のためだが，

movea.w a0,a0

は，a0.wを符合拡張してa0.lに収め直す命令なので，nopの代わりにはならない。

問4

メモリ上に，つぎのような4バイトのデータ列があり，a0がその先頭の00_Hを指しているとする。

00_H 01_H 02_H 03_H

この状態から以下の命令列を続けて実行すると，メモリ内容はどう変化するか。また，最後まで実行した時点でa0はどこを指しているか。

```
move.b (a0)+,(a0)+
move.b (a0)+,(a0)
move.b -(a0),(a0)+
move.b -(a0),-(a0)
move.b (a0)+,-(a0)
move.b (a0),-(a0)
```

【難易度低】

ポストインクリメントつき/プリデクリメントつきのアドレスレジスタ間接形式を変な組み合わせで使うと何が起こるか，というだけの問題だ。実際にプログラムを書いて走らせてみれば一目瞭然だからノーヒント。もちろん，卓上で答えが出せればそれに越したことはない。

問4解答

68000はソースオペランドの評価を終えてから，デスティネーションオペランドの評価に入る。したがって，

move.b (a0)+,(a0)+

は，厳密に，

1) a0の指すメモリから1バイト読み出す
2) a0を1進める
3) 1)で読み出した内容をa0の指すメモリに書き込む
4) a0を1進める

の順序で実行される。これを踏まえて問題文の命令列を追っていくと，最終的に元の4バイトは，

02_H 02_H 02_H 02_H

のように02_Hで埋められ，a0は先頭に戻ることがわかる。なお，途中にある，

move.b -(a0),(a0)+

と，

move.b (a0)+,-(a0)

の2命令はメモリ内容もa0の内容も変化させないので惑わされないように。

問5

あるプログラム中に，

```
movem.l d0-d7/a0-a6,-(sp)
bsr     foo
movem.l (sp)+,d0-d7/a0-a6
```

という部分があったとする。a7（＝sp）を除く全汎用レジスタをスタックに保存してからサブルーチンfooを呼び出し，戻ったら保存しておいたレジスタ内容を復帰する，というありがちな処理だ。ところが，このプログラムは，

movem.l (sp)+,d0-d7/a0-a6

の箇所で毎回バスエラーを起こすという。しかも，試しにサブルーチンfooをrtsだけの空サブルーチンにしても事態は改善されなかった。こうなる理由を思いつくかぎり挙げなさい。【難易度中〜高】

いちおうはデバッグ勘についての問題。また，68000についてのあるマイナーな知識の有無を問う問題でもある。

問5解答

サブルーチンfooは問題ない（らしい）ことが確認できたので，選択肢は案外少ない。ありがちなのはサブルーチンfooが実は実行されていなかった，というケースだろう。つまり，プログラムのほかの部分に誤りがあって，spに変なアドレスを入れたまま，

movem.l (sp)+,d0-d7/a0-a6

の位置に直接飛んできている可能性を疑いたい。

そうでなければ，movemのバグだ。デスティネーション側にレジスタリストがくるパターンのmovemには，実際よりも1回余計にメモリアクセスするという不具合がある。通常は問題にならないのだが，スタックをメインメモリの最高位に確保してあったりすると，その直後のメモリが実装されていない空間にアクセスしにいってバスエラーを起こす。

X68000＋Human68kの環境では，決まった大きさのスタックを確保する代わりに，自身に割り当てられたメモリブロックの末尾部分をスタックとして使用するようなプログラムを書くと，稀にこの不具合に引っ掛かる。もっとも，本来は引っ掛かるはずのプログラムでも，高位メモリをほかの用途（端的なところではRAMディスク）に使用していると，spが最高位アドレスを指すことがなくなるので，正常に動作“してしまう”。

リスト1に実験用のプログラムを用意した。RAMディスクを確保していない状態で走らせて，バスエラーが発生する様子を確認してみてほしい。それがすんだら，

リスト1

```
1:        .include doscall.mac
2:
3:        movea.l 8(a0),sp
4:        move.l  d0,-(sp)
5:        movem.l (sp)+,d0
6:
7:        DOS     _EXIT
```

movem.l(sp)+, d0

を，

move.l　(sp)+, d0

に変更し，movemではなくふつうのmoveならバスエラーが発生しないことも確認しておこう。

問 6

つぎの命令

mulu.w　d1,d0

の実行に要するクロックサイクル数の“最小値”を，必要なら実測のうえ，推定しなさい。

【難易度中～高】

“必要なら”なんていっているが，68000のマニュアルには最大値（70クロック）しか記されていないから，実測しないことには答えは出てこない。とはいえ，43億通りの組み合わせをすべて試すのは非現実的だ。68000が内部でどうやって乗算を実行しているのかを想像し，仮説を立て，検証することになる。

問 6 解答

結論からいうと，最小値は最大値の70クロックより32クロック分短い38クロックと推定される。根拠は以下のとおり。

muluの遅さから，68000は乗算をシフトと加算に置き換えて行っていると考えられる。以前，算術演算アルゴリズムを扱った回（1992年 9 月号）に示した乗算ルーチンと同様のアルゴリズムにより，2 進数レベルで筆算をシミュレートしているわけだ。ということは，muluの実行時間を変動させる要素と，マシン語で書いた乗算ルーチンのそれとのあいだには共通する部分があるに違いない。

リスト 2 は以前に示した32ビット×32ビット乗算ルーチンからの抜粋だ。これを見ると，ほとんどの命令はループ中で毎回実行されているが，唯一，7 行目の加算命令ひとつは，実行されたり，されなかったりしていることがわかる。実行されるのは，乗数中の注目しているビットが立っているときだ。このことから，muluも，乗数中の“立っているビットの数”が増えるにつれて，加算 1 回分ずつ，実行時間が長くなるだろうと推測できる。

試しに実行時間を計った結果が図 1 だ。被乗数d0は 0 に固定し，乗数d1を，

0000000000000000_B

0000000000000001_B

0000000000000011_B

0000000000000111_B

⋮

0111111111111111_B

1111111111111111_B

と変化させながら，それぞれ10万回掛け合わせるのに要した時間を計った（実際には100万回ずつ実行して，10で割っている）。計時にはIOCSコールONTIMEを利用し，空ループの時間は引いてある。マシンはクロック10MHzのX68000なので，得られた1/100秒単位の時間は，実行に要したクロック数と見なしてよい。

予想どおり，ほぼきれいな直線上に並んでおり，乗数中の立っているビット数が 1 増えると，実行時間は 2 クロック分増えることが見てとれる。最悪時の実行時間がマニュアルに示された70クロックを超えているなど，全体に理論値よりも遅くなっているのは，タイマ割り込みがかかっていることからくる誤差だ。図 1 ではやっていないが，その分を計算に入れて補正すれば，理論値とも一致する。ただ，途中 1 カ所，計測誤差そのほかでは説明のつかないブレが見られるあたり，68000も奥が深い（といって逃げる）。

今回は68000の比較的すみっこのほうの話題をクイズ形式でお届けしてみた。たぶん，次回は予告していた圧縮関係（はっきりいうとハフマン符合化）に戻れる，と，思う（ただ，いったんケチのついた原稿を完成させるのって，労力が要るんだよね）。

では，最後に，まだゆとりがある人にもう 1 問。

問 7

つぎの命令

divu.w　d1,d0　　（ただしd1≠0）

の実行に要するクロックサイクル数の“最小値”を，必要なら実測のうえ，推定しなさい。　【難易度中】

問題の答え自体は“思い切り引っ掛け”なのだが，できれば真面目にdivuの動作を推測してみてほしい。

リスト2

```
1:          move.l   d0,d2          *d2 = 被乗数
2:          moveq.l  #0,d0          *d0 = 積格納先
3:          moveq.l  #32-1,d3
4: loop:    add.l    d0,d0          *中間結果を左シフト
5:          add.l    d1,d1          *乗数の上位 1 ビットが
6:          bcc      next           *  0なら何もしない
7:          add.l    d2,d0          *  1なら被乗数を足し込む
8: next:    dbra     d3,loop        *32ビット分繰り返す
```

表1　mulu　10万回の実行時間

吾輩はX68000である
［第26回］

DMAしようよ

Izumi Daisuke
泉　大介

DMA転送，アレイチェーン，リンクアレイチェーン
DMACにはいろいろな使い方がある
用途に応じて必要なものを選択しよう

illustration：H.Yamada

こういうのを晴天の霹靂というのだろうか。最近Macintoshに，Centris660AVとQuadra840AVというAVシリーズが投入された。このマシンは，吾輩が今日まで心ひそかに描いていた夢を一気に打ち砕いてしまうほどの衝撃をもたらした。吾輩が描いていた夢というのはほかでもない。うちの御仁が折りに触れては口にしていた，「吾輩を手にしたときの，あのワクワクするような高揚した気持ち」を，もう一度プレゼントしたいというささやかな願いである。

図1　_DMAMOVE使用時のレジスタ設定

図2　IOCSコール8AHで実行されるプログラム

```
00FFA956    cmp.l    #$0000FF00,D2    * FF00Hと比較
00FFA95C    bcs.s    $00FFA974        * 小さければFFA974Hへ
00FFA95E    move.l   D2,-(A7)         * 転送バイト数を保存
00FFA960    move.l   #$0000FF00,D2    * FF00Hバイトを転送バイト数として
00FFA966    bsr.s    $00FFA974        * DMA転送
00FFA968    adda.l   D2,A1
00FFA96A    move.l   D2,D0
00FFA96C    move.l   (A7)+,D2         * D2を取り出す
00FFA96E    sub.l    D0,D2            * 転送バイト数をD2から引く
00FFA970    bne.s    $00FFA956        * 0でなければFFA956へ
00FFA972    rts
```

御仁の様子を見ていると，最近一段とMac贔屓が強くなってきているようで寂しい。QuickTimeが登場したときの御仁の興奮のしようったらなかった。たかだか160×120ドットの小さな絵が動いている程度のことで，なにをそんなに興奮することがあるのか，という意見もあろう。また，この程度のことならX68000でいつでも簡単に実現できる，という意見もあろう。御仁が興奮したのは小さな絵が動いていたからではなかった。それはいわば，新しいテクノロジーの萌芽に立ち合っているような，そんな気持ちだったに違いない。絵のサイズはハードウェアとソフトウェアの革新によっていずれ問題とならなくなる，実用性はどこかの誰かが考えてくれるだろう。そんな素のままのテクノロジーを標準機能としてOSに組み込んでしまう不敵さ。このマシンとつき合っていると，なにやら面白ことが起きそうだ，楽しい夢を見せてくれそうだ，という予感。そう，それは誕生した当時の吾輩が，全身にみなぎらせていたあの魅力だ。

ここは一発，起死回生のリーチをかけなければならない。なんとか御仁をワクワクさせる手はないものか。そんなことにアレコレ考えを巡らせ，これなら御仁の興味を掻き立てられそうだという仕様を練ってみた。

現在吾輩が最も得意としている分野はなんだろうか。3Dのレンダリングやモデリング，それにDōGAやMATIERを始めとするグラフィック関係である。いずれも実力のあるソフトウェア揃いだ。このあたりを敷衍（ふえん）させていくのが面白そうである。そもそも吾輩は16ビットカラーのグラフィックとFM8重和音，AD PCMといった装備を備えて誕生したAV指向のマシンだ。かく生を享けたからには，その道をまっとうするのが筋というもの。これに映像メディアを加え，テレビとパソコンの融合を図るのだ。奇しくもこれは，初代X1が目指した道で

もある。この分野にはAMIGAという先駆者がいるものの，幸いこいつはマイノリティ。十分駆逐でき，映像をやるならX68000といわれるステータスを確立できそうだ。

◆DMAその3通りの使い方

では，大容量のデータ処理を得意とするDMAを引き続き攻める。メモリの動作，原理などを順に説明し，前回は吾輩のIOCSに用意されているDMAルーチンを使って，実際にデータを転送してみた。CPUでのデータ転送プログラムも掲げておいたので，そのスピードの差を確かめていただけたことと思う。今回は，IOCSに用意されているそのほか2つのDMAルーチンを説明する。

まずはIOCSコール$8B_H$の_DMAMOV_Aについてである。前回使用したIOCSコール$8A_H$の_ DMAMOVEは，図1のようにD2.lに転送したいデータのバイト数を直接セットして使用する仕様になっていた。このため実に安直にDMAを使用できるのだが，いかにDMAといえども万能ではない。仕様をご覧になればおわかりいただけると思うが，吾輩が内蔵しているDMAコントローラHD63450では，一度に転送できるデータ数に制限が設けられている。図2はIOCSコール$8A_H$で実際になにが行われているのかを逆アセンブルしたものだが，これを眺めると指定された転送サイズを$FF00_H$バイトごとに分けてDMAを使っているのがわかる。実はHD63450のデータ転送カウンタは16ビットしかないのである。_DMAMOVEルーチンはユーザーの便宜を考えて，転送したいデータ数を32ビットで指定できるようにうまく計らってくれているわけだ。

このデータ転送カウンタの実体がよくわかるのが，_DMAMOV_Aルーチンである。_DMAMOVEルーチンではプログラムによって$FF00_H$個以下のデータ転送を繰り返すようになっていたが，_DMAMOV_Aルーチンでは図3のようなチェーンテーブルと呼ばれる表を使って，ユーザーが直接指示を与えるようになっている。このテーブルには，データ転送したいアドレスとそのとき転送するデータのバイト数が組みになった転送情報が延々と並んでおり，テーブルに何個の情報があるかは別途指定する。表の転送バイト数の部分に注目していただきたい。アドレス部分の半分の大きさ，すなわち16ビットになっているのを確認していただけるだろう。

この情報テーブルを使用するIOCSコール$8B_H$の_DMAMOV_Aの使い方は，図4のようになっている。まずD1.bには転送モードを設定する。読み方を説明しておこう。これは諸兄お馴染みのmove命令になぞらえて見ていただければと思って採用した表記で，たとえば，

D1.b＝00000101_B

だとすると，

move.b　(a1)＋, (a2)＋

という動作を指示したことになる。ここで注意していただきたいのは，_ DMAMOV_ Aルーチンではa1.lはテ

図3　_DMA_MOV_Aが使用するチェーンテーブル

図4　_DMAMOV_A使用時のレジスタ設定

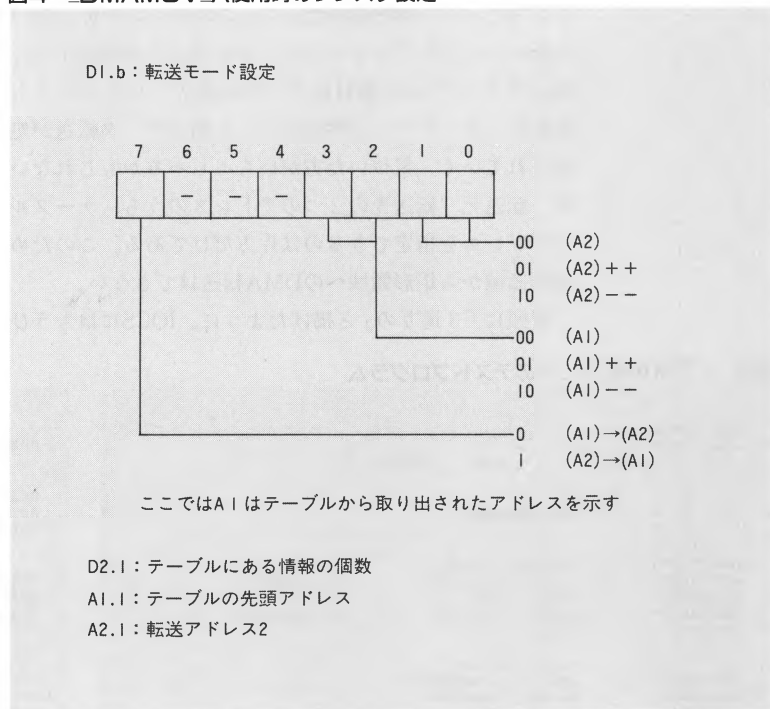
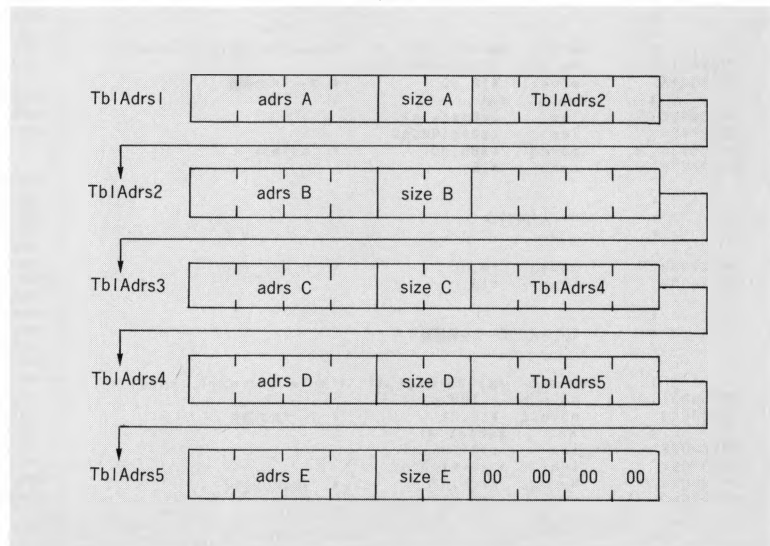



図5　_DMAMOV_Lでのテーブルの組成

ーブルの先頭アドレスを示しているということである。このため，上の説明は本当は正しくない。a1の部分を「A1.lで示されるテーブルから取り出されたアドレス」に読み替えていただきたい。次にD2.lには，テーブルに格納したデータの個数を設定する。アドレスとバイト数をひと組として，データがいくつ登録されているかを指定するのだ。A1.lにはテーブルの先頭アドレスを，A2.lにはデータを転送するアドレスを指定する。

このようにレジスタをセットして_DMAMOV_Aルーチンを使用すると，DMACに必要なデータがセットされたあとDMAが起動される。すなわちCPUは仮死状態に陥り，DMACがバスを乗っ取りデータ転送を開始するのだ。たとえばD1.bで「(a1)+，(a2)+」が指示されているとすると，DMACはテーブルから最初のデータ転送アドレスとバイト数を取り出して，このアドレスからA2.lにセットしたアドレスへシコシコとデータを転送する。続いてテーブルの2番目のデータ転送アドレスとバイト数を取り出して……，3番目の……と順次データ転送が処理されていく。気づいた方がいらっしゃるかもしれないが，転送元，転送先の2つのアドレスのうち，テーブルでアドレスを指定できるのは片方だけである。このため，矩形領域から矩形領域へのDMA転送はできない。

冒頭に「3通りの」と掲げたように，IOCSにはもうひとつDMA転送が用意されている。IOCSコール$8C_H$のDMAMOV_Lと名づけられているルーチンがそれだ。こちらはリンクアレイチェーンと呼ばれる方法を使うためのもので，_DMAMOV_Aとはテーブルの構成が異なっている。図5を見ていただきたい。データ転送を行うアドレスとそのサイズに続いて，次のテーブルの所在を示すデータが付加されているのを確認していただけるだろう。_DMAMOV_Lでは，DMACはこのリンクをたどりながら次々とデータを転送していく。テーブルのリンクの最後を示すデータは00000000_Hである。

_DMAMOV_Lは，条件によっては転送しなくてもいいデータがある場合に便利に使える。リンクを書き換えるだけで，データ転送の一部を簡単にスキップさせることができるからだ。逆に，データを転送する領域がきっちり決まっている場合は，リンクを参照する分余計な手間がかかるし，テーブルも大きくなるといったデメリットがある。用途によって使い分けたいIOCSルーチンだ。

◆DMAしてみる

実例をお目にかけよう。図6は例によって例のごとくデバッガによるプログラムである。冒頭でZ0というデバッガ内の変数にプログラムを作成するアドレスをセット

図6 _DMAMOV_Aのテストプログラム

```
-z0=200000
-an .z0
        †_exit          equ     $ff00

        *
        *       元となる図形を描く
        *

00200000        move.w  #16,d1
00200004        moveq   #$10,d0         * _crtmod
00200006        trap    #15
00200008        moveq   #$90,d0         * _g_clr_on
0020000A        trap    #15

        †       lea     box,a1
0020000C        lea     .z0+$50,a1
00200012        moveq   #$ba,d0         * _fill
00200014        trap    #15

        *
        *       バッファにデータを転送
        *

        †       move.b  #%0000_0101,d1  * move (ards1)+,(adrs2)+
00200016        move.b  #_00000101,d1
0020001A        move.l  #10,d2          * テーブルの個数
        †       lea     table,a1
00200020        lea     .z0+$5a,a1
00200026        lea     .z0+$100,a2
0020002C        moveq   #$8b,d0         * _mdamov_a
0020002E        trap    #15

        *
        *       キー入力を待つ
        *

00200030        moveq   #0,d0           * _b_key_inp
00200032        trap    #15

        *
        *       取り込んだデータを再描画する
        *

        †       move.b  #%1000_0101,d1  * move (adrs2)+,(adrs1)+
00200034        move.b  #_10000101,d1
00200038        move.l  #10,d2          * テーブルの個数
        †       lea     table1,a1
0020003E        lea     .z0+$96,a1
00200044        lea     .z0+$100,a2
0020004A        moveq   #$8b,d0         * _dmamov_a
0020004C        trap    #15

        †       dc.w    _exit
0020004E        _exit

        box:
00200050        dc.w    0,0
00200054        dc.w    50,10
00200058        dc.w    11

        table:
0020005A        dc.l    $c00000
0020005E        dc.w    50*2
00200060        dc.l    $c00000+$800
00200064        dc.w    50*2
00200066        dc.l    $c00000+$1000
0020006A        dc.w    50*2
0020006C        dc.l    $c00000+$1800
00200070        dc.w    50*2
00200072        dc.l    $c00000+$2000
00200076        dc.w    50*2
00200078        dc.l    $c00000+$2800
0020007C        dc.w    50*2
0020007E        dc.l    $c00000+$3000
00200082        dc.w    50*2
00200084        dc.l    $c00000+$3800
00200088        dc.w    50*2
0020008A        dc.l    $c00000+$4000
0020008E        dc.w    50*2
00200090        dc.l    $c00000+$4800
00200094        dc.w    50*2

        table1:
00200096        dc.l    $c40100
0020009A        dc.w    50*2
0020009C        dc.l    $c40100+$800
002000A0        dc.w    50*2
002000A2        dc.l    $c40100+$1000
002000A6        dc.w    50*2
002000A8        dc.l    $c40100+$1800
002000AC        dc.w    50*2
002000AE        dc.l    $c40100+$2000
002000B2        dc.w    50*2
002000B4        dc.l    $c40100+$2800
002000B8        dc.w    50*2
002000BA        dc.l    $c40100+$3000
002000BE        dc.w    50*2
002000C0        dc.l    $c40100+$3800
002000C4        dc.w    50*2
002000C6        dc.l    $c40100+$4000
002000CA        dc.w    50*2
002000CC        dc.l    $c40100+$4800
002000D0        dc.w    50*2
```

しているので，諸兄のメモリ事情に合わせて適当に書き直して実験してみていただきたい。また，アセンブラを使う諸兄は†印をつけた行のほうを入力して，該当するデバッガ用の行は無視されたい。

まずこのプログラムでは，画面を768×512ドット16色モードにして，画面の左上に50×10ドットのシアンの四角形を描画する。続いてこれを，(0,0)-(49,0)，(0,1)-(49,1)，……，(0,9)-(49,9)の10の部分に分割して，アドレスZ0＋100_H(ここでは200100_H)へDMA転送して格納するようになっている。大きさの決まった矩形領域を転送するため_DMAMOV_Aを使っている。リストの最後に何行も並んでいるdcの山がそのデータだ。

このままではなにが起きたのかよくわからないと思うので，おまけのプログラムを付加しておいた。DMA転送が終わったあとプログラムはキー入力待ちになるので，リターンキーかなにかを押してみていただきたい。諸兄がなにかキーを押すと，格納したデータを今度は逆に画面の別の座標へと吐き出す。DMAの転送方向を逆に設定し，別のテーブルを参照して転送アドレスを変更している点に注意していただきたい。この程度のサイズのデータならば，CPUで転送してもそれほど手間ではない。前回ほどの感動はないかもしれないが，そこはDMAの使い方，ということで勘弁していただきたい。

ここまで2つの転送アドレスをの両方を増加させる例をお目にかけてきたので，最後に，片方を減少させる例をご覧に入れよう（図7）。これはIOCSコールBD_HのSYMBOLというルーチンを使ってグラフィック画面に表示した文字を，DMA転送を使って裏返しにしてみようという例である。転送元アドレスを増加させる一方で，転送先のアドレスを減少させてやれば，左にあったものが右に，右にあったものが左に転送されることになる。

このプログラムでは_DMAMOVEを使って，G-RAMからG-RAMへ1ラインずつデータを裏返しにしながら転送している。このための転送元アドレスと転送先アドレスは配列にして格納してあるのだが，インパクトのある大きな文字を，と考えて24ドットフォントを倍に拡大したため，配列は96行にも及んでしまった。これではあまりにも格好悪いので，fromTable, toTableとラベルを振ってある場所にプログラム中で生成することにした。ここで注意していただきたいのは，IOCSコールに用意されているDMA転送ルーチンは1バイト単位のデータ転送しか行わないことと，GRAMの構造である。GRAMは常に1ピクセル2バイトで構成されるため，これを単純に裏返し転送したのではパレットコードの上位バイトと下位バイトが逆転してしまうことになるのだ。

ここでは16色モードを使用しているので，転送先のアドレスを1バイトずらすことで解決している。同様のことを65536色モードで実行したいという諸兄は，DMAのレジスタに直接データをセットして，1ワードごとにデータを転送する必要がある点に注意していただきたい。具体的な方法は，次回ご紹介することにしよう。まずは，IOCSルーチンで遊んでみていただきたい。

図7　_DMAMOVのテストプログラム

```
-z0=200000
-an .z0
        †_exit          equ     $ff00

        *
        *       元となる図形を描く
        *

00200000        move.w  #16,d1
00200004        moveq   #$10,d0         * _crtmod
00200006        trap    #15
00200008        moveq   #$90,d0         * _g_clr_on
0020000A        trap    #15

        †       lea     str,a1
0020000C        lea     .z0+$72,a1
00200012        moveq   #$bd,d0         * _symbol
00200014        trap    #15

        *
        *       DMA転送用のテーブルを作成する
        *

        †       lea     fromTable,a0
00200016        lea     $200100,a0
0020001C        move.l  #$c00000,d1
00200022        moveq   #47,d2
        makeFromTbl:
00200024        move.l  d1,(a0)+
00200026        add.l   #$800,d1
        †       dbra    d2,makeFromTbl
0020002C        dbra    d2,.z0+$24

        †       lea     toTable,a0
00200030        lea     $200200,a0
00200036        move.l  #$c50000+864,d1
0020003C        moveq   #47,d2
        makeToTbl:
0020003E        move.l  d1,(a0)+
00200040        add.l   #$800,d1
        †       dbra    d2,makeToTbl
00200046        dbra    d2,.z0+$3e

        *
        *       キー入力を待つ
        *
```

```
0020004A        moveq   #0,d0           * _b_key_inp
0020004C        trap    #15

        *
        *       データを転送
        *

        †       lea     fromTable,a3    * 転送元アドレスの格納アドレス
0020004E        lea     $200100,a3
        †       lea     toTable,a4      * 転送先アドレスの格納アドレス
00200054        lea     $200200,a4
0020005A        moveq   #48-1,d3        * データカウンタ

0020005C        move.l  #864,d2
        †       moveq   #%0000_0110,d1  * move (ardsl)+,(adrs2)-
00200062        moveq   #_00000110,d1
        loop:
00200064        movea.l (a3)+,a1        * 転送元アドレス
00200066        movea.l (a4)+,a2        * 転送先アドレス
00200068        moveq   #$8a,d0         * _dmamove
0020006A        trap    #15
        †       dbra    d3,loop
0020006C        dbra    d3,.z0+$64

        †       dc.w    _exit
00200070        _exit

        str:
00200072        dc.w    0,0
        †       dc.l    string
00200076        dc.l    .z0+$80
0020007A        dc.b    2,2
0020007C        dc.w    11
0020007E        dc.b    2,0

        string:
00200080        dc.b    "愛のある美しい文字",0

        †       .even
        †fromTable:
        †       ds.b    192

        †toTable:
        †       ds.b    192
```

これからのDōGA

プロジェクトチームDōGA
かまた ゆたか

DōGAは，これから先，どこへ向かうのか？ その問題を真剣に検討してみました。そのひとつの結論は，OUCC&KMCという従来の体制を改め，広く一般からDōGAの参加者を求めるということです。

はじめに

8月号のこの連載のなかで，“大阪デザイナー専門学校では，DōGAで修業すると単位がもらえる”というようなことを書きましたが，あれはもちろん冗談ですよ。数名の方が学校に問い合わせたそうですが，決して本気にしないでください。確かに，単位が足りない方にとっては切実な問題でしょうが，間違っても信じないでください。だいたい，この連載に書いてあることなんて，8割方デタラメなんですから……というのも，もちろん冗談ですよ。

さて，DōGAも活動を始めてはや7年。CGAシステムの開発，アマチュアCGAコンテストの主催など，さまざまな活動を行ってきました。ここ1年のあいだでも，CGAマガジンの創刊や一部法人化など，新しい試みも行っています。

しかしながら，長い歳月のあいだに状況は変化してきています。最初は大学のコンピュータクラブ間の共同プロジェクトにすぎなかったものが，いまでは全国的な活動となっています。また，設立当時のスタッフも，もうほとんどいなくなり，現在の活動を支えているのはあとから参加した人たちです。さらに，一部を法人化したことで，DōGAがアマチュアなのか営利団体なのかよくわからなくなり，ひいてはDōGAの目的も不明瞭になってしまった感があります。

そこで今回は，もう一度皆さんと一緒にDōGAの目的を確認すると同時に，これからの活動の目標を定めたいと思います。

DōGAの目的

DōGA設立時に掲げた目的は,「手軽でパーソナルな映像表現としてのCGアニメーションの普及」でした。そして，今後の目的は,「手軽でパーソナルな映像表現としてのCGアニメーションの普及」です。……つまり，まったく何も変わりません。当たり前ですよね。団体の目的が，コロコロ変わっちゃ困りますもんね。

この目的を実現するためには，まずは，CGA制作環境を整えることから行わなければいけません。CGA制作環境というのは，プログラムだけを意味するものではなく，よいマニュアルの制作，有効なデータベースの蓄積，またそれらを効率よく配布することも重要です。

さらに，より多くの人にCGA制作を始めようという気を起こさせる環境という意味で，作品の発表の場，人の作品を見る場を設けるのも大切です。CGAマガジンやこの連載もその一環といえるでしょう。

将来，パーソナルレベルでのCGAによる映像制作がごく一般的になり，日本の映像文化となる日まで，DōGAは戦い続けるのです。

具体的な活動内容

それでもって，その目的を実現するための具体的な活動内容は……，これも基本的に従来の活動を踏襲します(なんか，どこぞの連立政権みたいだなぁ)。しかし，現状に合わせて，いくつか追加・変更を加えます。

1) 本格的に，CGAシステムver.3の開発に入る。
2) 阪大・京大のコンピュータクラブ員以外の参加を募る。
3) データベースの蓄積を促進し，有効利用していく。
4) CGAコンテストは今後も継続し，定期的な上映会を全国で行う。
5) CGAマガジンは，6号(平成6年4月発行予定)までは現在の形態で発行し，以後については見直す。

特に重要なのが，2つめです。ご存じのように，DōGAは，大阪大学コンピュータクラブ(OUCC)と京大マイコンクラブ(KMC)が中心になって活動を行ってきました。しかしながら，いまとなっては，特にこの形態にこだわる必要はないと思うわけです。

設立した当時は，もちろんDōGAもまったく無名で，その活動の実現性もきわめて難しい状況にありました。しかし，現時点では，DōGAの活動趣旨，活動内容は，Oh!Xなどを通じて広く知れわたるところとなり，事実，現在は上記の2つのクラブ以外の大学生，社会人，高校生がスタッフとして活躍しています。

別に大学でなくても，また，コンピュータクラブでなくても結構です。また，プログラム開発だけが参加の方法ではありません。詳しくは，あとで解説します。

DōGAの活動に参加するにあたって

まず最初に，DōGAへ参加することのメリットについてお話しします。DōGAに参加すれば，CGAコンテストのビデオやCGAマガジンがただで貰えるとか，年に1回の大阪での懇親会に招待される……などといったくだらない理由を挙げるつもりはありません。現在のスタッフたちも，そんなことを目当てに活動しているのではありません。

皆さんは，DōGAの活動がよいことだと思われますか？　また，やるべきことだと思われますか？　もし，答えがYESなら，あなたも参加するべきなのです。

この際はっきりいいましょう。DōGAに参加しても，なんらメリットはありません。ただ，それがあなたにとって，またみんなにとって，よいことであり，やるべきことならば，何かメリットなど必要あるのでしょうか？

また，もうひとつの問題点についても先に話しておきましょう。いままでも，ほかの団体に声をかけたとき，「いや～，うちはゲームばっかりやってて，BASICでプログラム組める人間すらろくにいないから無理ですよ～」という返事が返ってくることが何度かありました。しかし，それは違うと思います。BASICもできないからプログラム開発できないのではなく，プログラム開発する必要性がないから，プログラムできるようにならないのではないでしょうか。

まず，「DōGAの活動に参加して，自分たちのプログラムを全国の皆さんに使ってもらおう」という目標をもち，「誰が，いつまでに，どんなプログラムを開発する」という計画を作れば，ちゃんとプログラムできるようになるはずです。

夢があるから，目標ができる。目標があるから，計画する。計画があるから，実行する。実行があるから，成果があがる。成果があるから，成長する。成長するから，さらに夢ができる……。これが「夢実現サイクル」です。

逆に，夢がなければ，目標もない。計画もない。実行もしない。成果は何もない……。こんなサイクルに入ってしまっているのではないでしょうか？

私は，極端な話，DōGAのプログラム開発に参加したが，どのプログラムも完成には至らなかった……なんてことになっても，別にかまわないと思います。たとえDōGAのプログラムが完成しなくても，その過程で，プログラム開発の楽しみがわかったり，C言語の知識が身についたりする人が必ず現れるはずです。あなたの努力は，確実に成果となるのです。もう「ゲームだけのクラブ」ではありません。

私は，最初に「DōGAに参加しても，なんらメリットはない」と書きましたが，本当はそんなこと思ってはいません。

参加内容

「よし，やってみよう」「興味がある，面白そうだ」と思っても，具体的に自分のクラブが参加するとなると，何をするのだろうか，何ができるのだろうかという疑問が出てくると思います。もう少し，具体的に考えていきましょう。

主なところでは，

1)　プログラム開発
2)　データベースの作成
3)　上映会の企画
4)　運営への参加

などが考えられます。

1)　プログラム開発

もちろんCGAシステムver.3の開発です。次のページのコラムにありますように，その前に共通規格から検討を始めないといけません。スケジュールとしては，年内に共通規格ver.3を決定し，それに合わせたシステム概要を来年春までに決めます(ライブラリの概要含む)。ですから本格的な開発は，来年春からとなります。それから約1年間で一応動くものを発表したいと思います。

プログラムの開発といっても，ベースとなるライブラリの開発から，通常のツール，大きなツールのモジュールなどがありますので，あまり大きなツールに自信のない方は，モジュールの開発なんかが参加しやすいと思います。

モジュールというのは，Z'sSTAFFの拡張機能(Z'sEX)のようなものです。たとえば新しいモデラは，メインに現在のCAD.XやFFE.Xのようなツールがあって，そこからいろいろな形状を作成するモジュール(現在のTAMEN.XやMIRR.Xなど)や変形するモジュール(EXPOINT.Xなど)を呼び出して使用するという使い方になると予想されます。

どのクラブがどのようなツールやモジュールを作るかという問題は，各クラブの希望をもとに，当方で調整していきたいと思います。

なら来年まですることない……というわけではありません。まず，当然のことながら，Cでプログラムできる人を養成する必要があります。すでにCができる方は，練習がてら現在のCGAシステムのツールを作ってみてください。ver.3の開発に入るといっても，しばらくの間はver.2と併用せざるを得ません。特にアイデアがないという方は，当方でなんぼでも用意させていただきます。さらに，パワーユーザーの方には，ライブラリの開発の段階から参加していただきたいと思います。

また，何もX68000シリーズユーザーしか参加できないというわけではありません。ご存じのように，REND.Xなど一部のプログラムはいろいろな機種に移植され活躍しています。DōGA内でも，モデリングはX68000で，レンダリングは486マシンで，という形が定着しています。ほかのマシンのユーザーの方も，そのマシンのCGA環境を改善するため，移植やオリジナルプログラムの開発に参加してください。

とりあえず，プログラム開発に参加を希望される方は，何か自作のプログラムを送ってください。

2) データベースの作成

DōGAの活動には参加したいが，プログラム開発はちょっとという方でも，このデータベースの作成は大丈夫です。

何かちょっとした作品を作ろうとしても，背景などたくさんの形状データを作らなければいけないというのは，

新CGA共通規格に関する考察

CGAシステムの ver.3 を開発しようといい出してから，もう1～2年が経っている……が，まったく進んでいない。

プログラム開発力が落ちているわけではない。むしろ，ver.2のバージョンアップは活発だ。しかし，それではver.3に近づかない。ver.3とは何か？　それは，データフォーマット(CGA共通規格）が異なったシステムだ。だから，新しいCGA共通規格が決まらなければ，決してver.3はできない。

そもそも，CGA共通規格とは何だろうか。マニュアルにも載ってはいるが読んだ人など少ないだろう。しかし，CGA共通規格は，CGAシステムだけでなく，DōGA自身にとっても，たいへん重要な命綱となっている。

CGAシステムの最大の特徴は，1つの大きなプログラムではなく，非常に多くの小さなツールから構成されていることだ。そして，これらのツールは，すべてCGA共通規格で定められたフォーマットのデータを入出力する。このことによって，同時に，たくさんの人が，別の場所でツールの開発を行うことができる。まさに，現在のDōGAの活動に適した形態だ。

ちょうどDōGAの設立日に，OUCC，KMC，大阪府立大学が一緒にパーソナルベースのCGAシステムを開発しようと話し合ったとき，最大の問題点がそこにあった。1つのプログラムをどうやって3つのコンピュータクラブで開発するのか？

入出力するデータのフォーマットさえ先に決めておけば，それぞれが自由に分業することができるはずだ。私のこの提案によって，DōGAの設立と同時に共通規格が生み出された。

共通規格のメリットは，ほかにもある。小さなツールの集合体となっているため，複合的なバグの発生がほとんどなく，バグ出しがしやすい。また，バージョンアップもやりやすい。新しく機能(ツール)を加えるのも容易だ。

形を定める形状ファイル，動きを定めるフレームファイルを用意したことで，モデラとレンダラを分離することにも成功した。こうすることによって，複数のモデラとレンダラをその場に合わせて，自由に選択できるようになる。もっとも，現在のCGAシステムは，モデラ関連のツールは複数あっても，レンダラは1つしかないのであまり有効に使われていない。しかし，この発想は，その後に発表されたピクサー社の『レンダマン インターフェイス』を先取りしている。

この共通規格が発表されて，もう5年になるが，さすがに問題点が出てきた。一部記述がおかしいと思われるところ(バンプマッピングの記述など)もあるし，その5年間に進歩してきたCGの技術は取り入れられていない(オクトゥリー分割など)。また，もっと切実な問題として，時間軸方向に，形状や色が変更しにくいといった点がある。

考えてみれば当たり前で，この発展が著しい業界において，5年間も通用するような規格を定めることができたほうが，むしろ不思議といえる。たぶん，CGの知識も足りない素人だからこそ断行できたのではないだろうか。

そこで，これらの問題点を見直し，さらにこれから5年間の動向を予想して，新しい規格を作っていきたい。……作っていきたいのはやまやまだが，責任重大だし，非常に難しい。とっても難しい。う～ん，どうしよう。困ったな。う～ん，う～ん……。

DōGAでも，Ko-WINDOWで有名な小林を中心に，共通規格検討委員会を設置して草案作りを行っている。表現力豊かであり，フォーマットとしてはシンプルで，実用性を重要視する……そんな規格を目指している。さらに，多くのアマチュアが開発に参加できるような配慮も忘れるわけにはいかない。

ということで，みなさんのご意見を伺いたいと思います(こればっかしやな)。特に，プロのCG屋さん，ワークステーションのCGシステムを使用経験のある方，その方面の研究者など，専門的な知識を持っている方，どうかアドバイスをください。また，共通規格検討委員会に参加したいという方も募集しています。

そこで，ちょうど共通規格に関するご意見のお手紙が来ていますので，それを紹介して，DōGAでの具体的な議論のほんの一部を解説しましょう。

佐渡氏のご意見

Oh!X誌で触れられていたCGA共通規格の件ですが，まだ意見を述べることはできるのでしょうか。

実は，画像ファイルに半透明情報も追加してほしいのです。多段階半透明合成が可能になれば，アンチエイリアシング済みの画像を合成しても，妙な輪郭が浮き出ることはなくなるでしょう。これにより，背景作成や画像作成を繰り返すのがとても楽になると思うのです。表現力も上がるでしょう。

ごもっともなご意見です。では，もう少し具体的に考えてみましょう。現在のCGAシステムの画像ファイルのフォーマットでは，RGBが各5ビットで，残りの1ビットを半透明情報として使っています。しかし，1ビットでは，完全に透明か，完全に不透明かのどちらかしか表現できません。では何ビット必要でしょうか？

アンチエイリアシング2倍程度なら，4段階の濃淡ですから2ビットでもかまいません。将来的には4～5ビット欲しいところです。

しかし，たとえば5ビットにすると，大きな問題が発生します。RGB各5ビットに半透明が5ビット，ということは1ドットが20ビット。これは，コンピュータで扱うにはきわめて中途半端な値です。とてもじゃないですが，そんな規格は採用できません。

同時に考えなければいけないのが，RGBの色数です。現在の3万2千色では，マッハバンドが見えてしまいます。業界標準は1600万色，つまりRGB各8ビット，計24ビットとなっています。というと，「え～，X68000では，1600万色も出ないから無意味だよ～」という人がいるでしょうが，それは正しくありません。

まず，ver.3は5年間仕様変更せずに使用できるものを目標に設定すべきです。その場合，これから5年間，X68000が現在のままのスペックであるはずがありません。1～2年のうちに1600万色になると考えるべきです。また，3万5千色のデータから1600万色の画像を表示することはできませんが，1600万色のデータから，3万5千色の画像を表示することは簡単です。

とすると，RGB各8ビット，それに半透明ビットが4ビット，計28ビット。これでもやっぱり中途半端なビット数です。それでは，透明ビットを8ビットにして計32ビット。これなら，ちょうど32ビットのCPUが1度に処理できる量ですから，とても扱いやすくなります。

しかし，透明ビットは8ビットも必要ありません。ならば，4ビットにしておいて，残る4ビットをほかの情報に使いましょう。マスクビット？　でもそれは1ビットで十分。合成時の優先情報？　そんなのいるかなぁ。それに3ビットで十分？　ランレングスの情報というのには，ちょっと足りないし……。

また，RGB各7ビット，半透明3ビットで計24ビットという案だってあり得ます。RGBが7ビットあれば，マッハバンドは出ない(人間の目に見える色数は200～300万色なので，1600万色は無駄)し，24ビットなら，ちょうど3バイトでそこそこ扱いやすい。でも7ビットというのは扱いにくいですよね。

さらに，これらビット数の問題は圧縮アルゴリズムとも関連してくる話です。そうなると，話はさらに複雑になってきます。

というように，単純に半透明情報を追加するといっても，いろいろ難しい問題を含んでいるわけです。ここは皆さんの知恵をお借りして，じっくり考えていきたいと思います。お手紙をお待ちしています。

現在のCGAの大きな問題です。そこで，これからのDōGAの重点活動項目に，汎用性の高い形状データの蓄積を加えたいと思います。

建築物，樹木，人，車，家具などさまざまなものが考えられるでしょう。CGAマガジン創刊号のようにF-1などのテーマごとのデータベースも非常に有効です。また，形状データに限らず，背景用，マッピング用画像データなども重要です。

「とはいっても，CGAマガジンに載っているようなすごい形状よう作らんし……」と思うかもしれませんが，それは違います。確かに，CAD.Xをはじめとするモデリング用のツールにある程度精通する必要はありますが，現在のパソコンのメモリ容量などを考えますと，背景用の形状などは，できるだけ面数の少ない，簡単なデータである必要があります。もっとも，少ない面でそれらしい形にするというのもテクニックがいるのですが，それは，数をこなしているうちに自然と身についてきます。

このデータベースの構築に参加するのは，大石さんのように大戦兵器集を個人でコツコツ制作されるのもよいし，チームでその街の建物を片っ端から一気に作るというのでも結構です。まず，テーマを決めて，ご連絡ください。データベースを構築するうえでの注意(スケールやアトリビュートの名前の付け方など)をお送りします。

3)　上映会の企画

全国各地から，上映会をしてほしいというお便りをいただいているのですが，当方では会場を手配するのが非常に難しいのが現状です。そこで，上記の活動と並行して上映会などを定期的に催してはいかがでしょうか。

定期的にといっても，年1～2回が限度でしょう。また，最も簡単な形態としては，学祭のイベントのひとつにしてしまうという手もあります(ただし，来場者が片寄ってしまう？)。

上映会の主催・運営は，基本的に皆さんが自主的に行う形式となります。当方から協力できることといえば，Oh!Xなどでの告知，液晶プロジェクタの貸し出しなどで詳しい内容が決まりましたら，早めにご連絡ください。

上映内容としましては，CGAコンテストのビデオと自分たちのオリジナル作品を上映するのが一般的になると思いますが，CGAに関する講演会とか，特別な上映会ということでしたら，当方のスタッフの派遣，または特別な編集なども行います。

4)　運営への参加

これは大阪近辺の場合に限られてしまいますが，OUCCやKMCと同じようにDōGAの運営に参加してみるのはいかがでしょうか？　なんか大変そうに思えますが，要するにプロジェクトルームに出入りするということです。

現在，DōGAの運営は，何か問題が生じると，そのときプロジェクトルームにいるメンバーで話し合いが行われ，解決します。もう少し，時間をかけて検討すべきことは，プロジェクトルームに掲示されたり，DōGAノート(プロジェクトルームに常時置いてある連絡帳)に書かれます。ですから，特に何をするわけでなくてもよいのですが，とにかくプロジェクトルームに存在する時間が長ければ，自然と運営への参加率は上がります。

また，ある程度独立したプロジェクトになると，プロジェクトルームに出入りしているスタッフのなかから適当な担当者が決められ，その担当者を中心に必要な人手を集めて活動します。たとえば，CGAマガジンは，MAX田口君を編集長とし，マンデル北尾君などによって制作されており，私はほとんどノータッチです（だから，今回のような「SHIFT」事件が可能となる)。CGAコンテストも第4回はFFE三保，5回は遊び人松井，6回はマリオ古本が担当しています。

クラブ単位で参加される場合，こういったプロジェクトをひとつ担当してもらうという可能性もあります。CGAマガジンの一部や，データベースの整理，管理，または何かのイベントの段取りということもあるでしょう。しかしながら，こういったことはある程度経験が必要です。まずはプロジェクトルームに出入りして，誰かの指示のもとで何かのプロジェクトに参加するなど，ある程度経験値稼ぎを経てからになると思います。

参加方法

1) 了承を得る

あなたがやってみようという気になったら，まず，クラブ内の賛同，了承を得てください。個人参加ならいざ知らず，団体参加の場合は当然ですね。

なにもクラブ全体が参加するという形式をとる必要はありません。クラブ内に「CGAプロジェクト分科会」を作って，賛同者だけで参加するのもよいでしょう。実は，OUCCやKMCもそれに近い形態をとっています。

2) 計画をする

上記の内容を参考に，どのような形で参加するのか考えてみましょう。プログラム開発ができるのは何名いるのか。データベースを作るなら，どういったものを作りたいか。

そして，いちばん大切なことは，それらをいつまでに完成させるか，つまり締め切りを設けることです。締め切りのない計画なんて，いつまでたっても完成しません。

また，団体としての代表者以外に，各担当者も決める必要があるでしょう。要するに，計画どおりいくようにみんなにハッパをかける人です。責任の所在がはっきりすることで，何がなんでもやらねばという気持ちになるわけです。

3)　申し込み用紙を作る

以下の例を参考に，申し込み用紙を制作し，DōGAまで送ってください。

1)　団体名，代表者，連絡先など
2)　参加内容ごとの項目（プログラム開発・データベース作成・上映その他）
・担当責任者名，連絡先など
・参加人数，氏名，経験（使用可能言語・CGA制作歴など）
・開発計画（どんなことを，いつまでにするかなど）
・やってみたいこと（プログラム例・データベース例など）
3)　その他（自由に）
・各担当責任者の自己紹介など
・DōGAに対する要望・サポート・疑問など

特集：ホームズの挑戦

7月号で行いました，「CGA事件簿」のクイズ「ホームズの挑戦」がずいぶん好評で，たくさんの解答が寄せられました。残念ながら，完璧な解答はなかったものの，なかなかの名推理から見当違いまで，また，ちゃんとホームズとワトスンの会話調になったお便りもたくさんあり，ずいぶん楽しませていただきました。
こんなネタをほっとくのはもったいないので，いくつか紹介しましょう。
なお，以下4名の方を準当選者とし，内緒のプレゼントをお送りします。つまり，正解に近いかどうかとか，厳正なる抽選などまったく無視して，単に面白ければよいという，とってもDōGAな選び方になってしまいました。あしからず。

解答というより“怪答”

MIGA！さん

この挑戦，受けて立つ!!
DōGAの諸君，そして，ホームズ＆ワトスン君，ご機嫌いかがかな。私は，自称・名探偵のMIGA！である。
7月号の「CGA講座」は，いつにもまして，楽しく読ませていただいた。「技術推理小説」という形式にも興味をそそられたが，それにもまして，「懸賞付きクイズ」とは……。私に対する挑戦であろうか？
しかし，「問題が難しいから……」などと，いささか甘くみられているようである。本来私は，カネにならない仕事は受けないことにしているのだが，この私にもプライドというものがある。私の名推理，とくとご覧いただこう。

>謎を解く鍵たち
その1「正解は，Graphic Galleryの写真のどれかになるわけです」という部分。
これだけで，答えは12通りに絞られる。
（かまた：そりゃそうだ）
その2「ウソをついた（説明を省略した）というか，……」という部分。
わざわざ，カッコまでつけて省略して強調している。今回の講座で解説がない写真はただ1つ。ならば，自然と答えは見えてくる……。
（かまた：なんか深読みしすぎてるような……）
その3「球体」と書かずに，「金属の球体」とワザワザ強調している点（P.88）。
なぜだろうか？　それは金属でなければ困るからにほかならない。
（かまた：いやぁ～，単に映り込みをするような表面がツルツルの物体というだけで，他意はないんですけど……）
以上のヒントから，答えは写真9の「マッピングしたにもかかわらずスペキュラーが発生している」写真であろう。CGAシステムのマニュアル，T-228・アトリビュート研究にも書いてあるように，スペキュラー成分を0にしてマッピングさせた画像と，スペキュラーだけを作画させた画像とを，TPILE.X の/M モードで合成する……しか，実現する方法はないはずだ。
（かまた：へ～，CGAシステムのマニュアルにちゃんと載ってたんか。知らなんだ。みんな，ていねいにマニュアル読んでんねんなぁ）
さらにつけ加えておくと，リスト1は
spc (0.0 0.30 0.00)
と，なっているが，これは，
spc (0.0 0.00 0.00)
でなければならないのでは？
（かまた：そんなことないで。強さ成分が0なんやから，ほかのパラメータはなんでもかまへんで）
しかしながら，いまになって8月号を読むと，「正解者がいない」と書いてあるじゃないか。だけど，このヒントを読んでみてもなんのことかサッパリパァー。ちょっとくやしいけど，答えを楽しみにしているからね。
8月号にはクイズがなかったのがちょっと残念。こういうふうに，読者が参加できるクイズは続けてほしいな。まぁ，何にしても，これからも応援し続けますのでがんばってください。

「CGA」事件簿や，クイズについては，またネタを考えておきます。なお，お手紙には「お願い：ハズレてても，さらしモンにしないで」とありましたが，ボツにするにはもったいないお手紙ですし，内容も十分正解といえるので，恥ずかしがらなくてよいでしょう。

努力とアイデア

藤井さん

>いろいろ考える
本文を何度も読み返しても，抜けているような部分は見つからない。写真12だけ背景が黒なのに気がついて，試しに作ってみると，ちゃんとできてしまい，作る過程でほかの画像もできてしまい，いよいよわからなくなってしまった。一体どこに問題があるのかわからない。
>問題が見つかる
わからないものはしょうがないと思い，しばらくほっておいた。2，3日後に何気なく写真を見るが，……やっぱりわからない。ただ，同じ写真が2枚あることが目についた。なぜ同じ写真が2枚あるのだろうか？
さらによく見ると，まったく同じ写真ではないことに気づいた。写真9には白いもの，つまりスペキュラーがある。きっとこれが問題の部分だと思った。あとはスペキュラーを出せれば事件は解決だ。
>いろいろやってみる
面の色（白）とスペキュラーの色が同じ色だからまずいのではないのかと思い，色（青）のついた球を作画させると，スペキュラーは出るのだが，周りのタイルの絵を映り込ませるためにマッピングを行うと，やっぱりスペキュラーはなくなってしまう。どうやら，この方法ではうまくいかないようだ。
>ついに完成
アトリビュートをいじっても駄目だったので，あとはマッピングのデータしかない。スペキュラーのようなものがあるマッピング用画像データを作ることにする。さすがに，ペイントソフトでスペキュラーを描くのではなく，マッピング用画像データを作画させるときに，点光源や背景の面を与えて，スペキュラーを発生させた。
これで作画をすれば完成。やっぱり，邪道だろうか。
（かまた：はい，邪道です。でも，邪道も好き……。）
>もう少し考える
上の方法ではいんちき臭いので，まっとうな方法も考えてみた。まずマッピングデータを作るときに背景を透明にしておく。これを単純に張りつけると，背景のところが張りついた球の部分が透明になって，タイルの部分しか表示されない。そこで，その球より，ほんの少し小さな青い球を同じ場所に置くようにすれば，見た目にはこの小さな球にマッピングされたように見え，しかもスペキュラーを出すことができるのだ。
こちらのほうが正解ではないでしょうか？

正解ではありませんが，いただいたお便りの中では，最も斬新な方法でした。しかし，これだと，映り込んでいるタイルの部分にスペキュラーが出ないという問題があります。

本格的推理小説

河内さん

>プロローグ
6月の雨の日の午後，私は1冊の雑誌とブランデーを持って，彼の部屋を訪れた。
「やあ，起きてるかい」
ソファーに寝ている彼に声をかけると，彼は大きなあくびをしながら起き上がった。彼は私の親友で探偵をしている。シャーロック・ホームズに憧れているそうだ。
私はブランデーを机の上に置き，持ってきた雑誌を彼に見せた。
「Oh!X 7月号だよ。その最後に書いてある問題を解いてもらいたいんだ」
彼はOh!Xを手に取り，読み始めた。私はそのあいだに紅茶をカップに注ぎ，近くの椅子に腰掛けて飲み出した。

その他

上記は，大学のコンピュータクラブを想定して書かれていますが，もちろんそれ以外の方でもかまいません。スタッフに高校生がいるぐらいですから，高校や各種専門学校でも参加に支障はありません。プログラム開発は無理だとしても，データベースの制作には，アニメーション研究会，美術部から映画研究会，SF研究会などでもまったく問題ないと思います。もちろん「ぼくらの掲示板」に載っているようなサークルでも結構です。

個人参加の場合，人数が多いと，DōGAからの対応が十

>２つの答え

私は，どうやら眠ってしまったらしい。時計を見るとここへ来てから２時間がたっていた。

「どうだい，何かわかったかい」

私は彼と向かい合わせに座りながら尋ねた。

「ああ，とりあえず君の考えを聞かせてくれないか，君なりに考えてみたんだろう」

彼はパイプの煙を吐きながら，少しけむたそうにいった。

「よし，じゃあ……」

私は彼の手からOh!Xを取り，「Graphic Gallery」のページを開いた。

「いいかい，まずどの写真が答えになるかを考えたんだ。写真１と２は問題となる写真だから違うだろう。12はこういうことをするための講座なんだから，これの説明をしないわけにはいかない。そして４と６と８は説明の写真だ。これを作るわけじゃない。５と10はマッピング用の画像だから違う。３はマニュアルのページ数まで書いているんだからこれも違う。９は説明してないから省こう」

「なぜだい？」

彼は右の眉を上げ，上目遣いにいった。

「この問題には“ウソをついた”とあるんだ。説明してないんだからウソをついたことにはならないだろう。続けるよ。残っているのは７と11だ。この２つのどちらかが答えだと思うんだ。だから２つとも作ってみたんだ。11のほうはうまくできたよ。７のほうは５の画像をそのままマッピングすると上下逆さまになるんだ。だから，rotやscalで長方形を逆さまにしたり，アトリビュートファイルのmapsizeを書き換えて画像を逆にしたりして，ちゃんとした向きにしてやる。この，ちゃんとした向きにする説明がなかったんだ。だから答えは写真７になる……，と思ったんだが，これじゃあ単純すぎる。だから君のところへ来たんだよ」

彼は煙を，フゥーッと大きく吐き出し，

「なるほど，確かに写真７という答えも悪くはない。しかし，写真８を見ればそのぐらいのことはわかる。いいかい，結論からいうと写真９が答えだよ」

「な，なぜだい，さっきいったように９は説明してないじゃないか」

「説明はちゃんとしているさ。見てごらん」

彼は88，89ページを開き“球体への映り込み”の部分を指さした。

「ここに書いてあるのは11の説明ではなく，９の説明なんだ。君は，ここに書かれているとおりにやって11を作ったんだろう。しかし９はできなかった。つまり，これは“説明を省略した”９の説明なんだ。わかったかい」

彼は，パイプを置いて隣の部屋へ向かった。

「こっちへ来てごらん」

>合成

この部屋にはX68030がある。彼は，１枚の画像を出した。それは11の画像だった。

「いいかい，ここまでは説明どおりだ。ここからが省略されている部分なんだ」

彼は続いてもう１枚の画像を出した。今度は周りは黒く，真ん中あたりが白くなっている画像だ。

「この画像とさっきの画像を合成させるんだ」

そういうと，彼はさささっとマウスを動かしTPILEを実行させ，できあがった画像を出した。それはまさしく写真９だった。

「どういうことだい，これは」

「やり方はマニュアルに書いてあるよ」

彼は横に置いてあるバインダーを私に渡した。

「T-228の下のところさ」

私はマニュアルを開き，いわれたところを読んでみた。そして，すべてを納得した。

「よし，じゃあ，さっそくブランデーで乾杯といこうじゃないか」

私はバインダーを渡して，持ってきたブランデーを取りに部屋を出た。

>エピローグ

私が彼の部屋の出ると雨はすっかり上がり，月が出ていた。私は傘を回しながら家路についた。アルコールが入っているのも手伝い，私はとてもいい気分で歩いていた……が，途中で，彼の部屋にOh!Xを忘れてきたことに気づいた。ちょっと迷ったが，「満開の電子ちゃん」を読んでなかったので，急いで取りに戻ることにした。

６月のある日の出来事だった。

まぁ，電子ちゃんを読まないわけにはいきませんよね。あと，岡村姉弟４コマと。あっ，CGAシステムの新しいマニュアルを作るとき，岡村さんに依頼しようかな……。

９の写真を説明してあるくだりは少しわかりにくいですが，なかなか達者な文章でした。

名推理

毛利さん

これは，私の勝手な推理です。とんでもない勘違いをしているとは思いますが，「応募者はいるが，正解者はいない」という言葉でひらめいたのですが，いかがなものでしょうか？

まあ，違っていたら笑い話として受け取ってください。

＊　　　＊

私は，いつものようにホームズの部屋を訪ねる。相変わらず机の上には資料や専門書が開いたままになっている。その横に，妙に整理された封筒を見つけた。20～30通はあるだろうか。

私はその封筒を１つ手に取った。どうやら「ホームズの挑戦」係への応募らしい。

ホームズ（以下**ホ**）：それは正解者のものだよ。

眠っているものと思ったホームズが，ソファから起き上がっていった。

ワトスン（以下**ワ**）：へぇ，結構多かったんだね。じゃあ，応募全体はかなり多かったんじゃないかい。全部で何通来たんだい？

それを聞くとホームズはニヤリと笑った。

ホ：それだけだよ。応募した時点で皆正解者だからね。

私は，彼のいっている意味がわからなかった。

ワ：それはともかく応募されたディスクがないようだが？

ホ：いやいや，最初からディスクなんかなかったのさ。

ワ：ディスクがないだって？　それじゃ正解の画像が見られないじゃないか。

ホ：だから，画像なんて必要ないんだよ。

ワ：しかし，正解は「Graphic Gallery」の写真のどれかになるっていってたじゃないか？

ホ：……僕が８月号でヒントを出したのを覚えているかい？

ワ：ああ，「人は探しているものを見つけると，探すのをやめてしまう」　それと，えーと……。

ホ：「応募者はいるが正解者はいない」

ワ：いったいどういう意味なんだい，意地悪しないで教えてくれないか？

ホ：そうだね……。

ホームズの顔が若返ったように見えた。

ホ：僕はいったはずだよ。「あそこには作者の努力とアイデアが映り込んでいたんだよ」って。

ワ：それが正解と関係あるのかい？

ホ：おおありだよ，答えは「努力と試行錯誤」さ。

ワ：「努力と試行錯誤」，テクニックは？

ホ：それはおのずと身についてくるものだよ。確かにテクニックが必要になるときもある。しかし最後にものをいうのは「努力と試行錯誤」なんだよ。やってみればわかるが，実はあの説明だけでどの画像も十分できるはずなんだよ。

ワ：それじゃあ，君は最初から読者に問題とだまして実際にCGを作らせて，自信をつけさせたってことかい？

ホ：そのとおり。センスとか才能は，最初から持っているものではない，自分で学んでいくことが重要だってことさ。この封筒はそれに気づいた人のものなんだ。

ホームズは，そういいながら，次の依頼書を手に取るのであった……。

あのクイズに，そんな深い意図があったとは，筆者自身知らなかった。いや～，まいった。

ということで，アンコールも多いことですし，「CGA事件簿」の企画はまたいずれ行いましょう。こんどは，ポアロ調だったりして。金田一も面白いかな。コロンボは，先にトリックがわかっちゃうからやりにくいかな。

分できない可能性が心配です。しかし，誰にも知られずに活動を続けるというのも悲しい気もするので，「ここにこんなヤツがいて，こんなことをしようとしているんだぞ」という意味で，上記のような申し込み用紙を送ってくだされぱありがたく思います。その際，DōGAに対する要望などありましたら，ご記入ください（必ず応えられるかは自信ないが……）。

参加申し込みの締め切りは10月末日とします。ただし，11月に学祭がある場合は年末で結構です。また，その締め切り以降でも随時受け付けます。……なら，締め切りの意味がないような気がしますが，そのとおりです。

しかしながら，締め切りがないと，先ほど書いたよう

夫婦でQ&A

Tさん(東京)：かまたさん，うさ子さん，ご結婚おめでとうございます。CGAマガジン3号の「SHIFTキーのやつ」は，よかったです。
うさ子：「SHIFTキーのやつ」って，何ですか？ 私たちの結婚と何か関係あるのでしょうか？
ゆたか：いや，ワシは何も聞いていないぞ。おい，ちょっと誰か，3号貸してみい。こらっ，3号はどこや〜。
（スタッフの陰謀で3号のマスターは，私の知らないところに隠されてしまった。「SHIFTキーのやつ」って，いったい何なんだぁー。自分で，3号買わないかんねやろうか？）
Iさん(神奈川)：電脳倶楽部などで知ったのですが，バックグラウンドでプログラムを動かしてくれる「BGDRV.X」なるものがあるそうですね。このプログラムで「REND.X」を動かすことができるでしょうか？ また，どうやって手に入れればよいでしょうか？
うさ子：バックグラウンドってなんですか？
ゆたか：地面の背後……つまり地下組織のことだよ(嘘)。
うさ子：ふ〜ん……。
ゆたか：REND.Xには，バックグラウンド用に画面出力をしない「-U」というオプションがありますが，どっちにしろIさんは初代機なので，遅すぎて使いものにならないと思います。また当方はBGDRV.Xを使ったことがありませんので，正確にはわかりませんが，いろいろ問題がありそうです。入手方法は，パソ通が一般的ですが，やっていない方はちょっと難しいでしょう。電脳倶楽部に問い合わせてみては？ また，読者の方で詳しい情報がありましたらご連絡ください。
うさ子：あの〜，「BGDRV」って地下組織の名前なの？ なんかぜんぜん意味がわかんないんですけど……。
Nさん(大阪)：電子ちゃんのような広告マンガはやらんのですか？ モデルは，うさ子さんの少女時代ということで，どうです？
うさ子：CGAシステムの広告してもしかたがありませんし，特にその計画はありません。
ゆたか：ところでうさ子の少女時代って，どんなんやったん？
うさ子：小学校に入ったころは，とっても泣き虫でした。ところが，うちとこに学校の番長みたいな子が入門してきて……。
ゆたか：そうそう，実家は空手の道場やもんね。お義父さんは空手7段の師範なんて，YAWARAみたい。
うさ子：私はぜんぜんできへんけど。……それで，それからは，その子が私に対してだけは，師範の娘ということで，ちゃんと「さん」づけで呼び，礼までするようになって，以後だれもいじめなくなってん。
ゆたか：なんか，マンガみたいな話やな。
Mさん（福岡）：CGAの連載は，たしかに内容が難しいです。が，しかし苦労して理解できたときの爽快さも捨て難いです。もちろん初心者入門向けの内容も期待しています。それから，アマチュアのDōGAは，解散しないでください。いずれは解散するのでしょうが（ゼネプロとガイナックスのように），できるだけ長くがんばってほしいです。理由は，アマチュアの団体がリードしていると，初心者が参加しやすいという，私なりの勝手な考えからです。
うさ子：本文のとおり，アマチュアのDōGAはこれからもがんばっていきます。解散なんてありませんからご安心ください。
ゆたか：そうそう，先月号に「解散か!?」なんて書いたのは，衆議院のパロディです。私としては，会社がつぶれてもアマチュアは残したいと思っています。
うさ子：ところで，初心者向けの連載の計画は，どうなってるの？
ゆたか：次回の「EPA2補講」も初心者向けですが，そのあとに「初心者でも，オリジナルの，ものすごいCGAが，簡単に作れる！」という夢の企画を構想中です。でも，この企画のためにはちょっと準備が必要なんですよ。だれかモデリングが得意な人に手伝ってもらわないと……。志願者は，「謎の別冊CGAマガジン係」までご連絡ください。
Kさん(北海道)：法人化ですか。倒産しないようがんばってください。私は，会社を解雇されちゃったもんで……。しゃれになってね〜。
ゆたか：冗談じゃないですね。その可能性は十分ありますから。うさ子，そうなるとお前にも苦労をかけるねぇ……。
うさ子：それはいわない約束でしょ。
村瀬さん（西宮)：CGAマガジン定期購読希望（5インチ）。できれば，創刊号から送っていただければうれしいのですが……。めんどうだといわずによろしくお願いします。あと半年（早くて）すれば，肉体奉仕をしますので……。
うさ子：発送担当者の方，いかがですか？
担当者：めんどうだ。第一，創刊号，2号の分のお金がないぞ。
ゆたか：しかたがないので，淡路まで取りにきていただければうれしいのですが……。めんどうだといわずによろしくお願いします。
うさ子：そういうことは，合格してからのほうがよいと思うんですけど……。受験がんばってください。
Hさん（三宮)：「笑え この野郎II」CGAとLIVEのMIX MEDIA！8月29日 三宮G STASIONにて開催！
ゆたか：そういうことは，早めにご連絡ください。
Nさん（帯広)：お忙しいところ恐れ入ります。FFEの動作についての質問です。一度セーブしたデータに対して，もう一度起動したのち，同じフレームナンバーを選択，修正を加えると，出力されるフレームソースが，

```
    { mov ( ¥div ( -3000, -500, 500, 800,
              1,    1,  10,  20,fno )¥ ……
```

となって，同じフレームナンバーが複数発生してしまいます。あとでエディタで直すということをやってますが，仕様でしょうか，それとも……。フレーム1を10回ほど選択すると，10個ほど発生します。
うさ子：わ〜い，バグだ，バグだ。
ゆたか：喜ばんでもええ。でも，当方で実際にやってみましたが，そのような現象は起こりませんでした。ただし，昔そういうバグが発生したことはあります。Nさんのバージョンが古いものだと考えられます。チェックしてみたFFE.Xは，Ver.1.54(93-04-30)とVer.1.55(93-05-22)です。
うさ子：FFE.Xのバージョンアップは非常に頻繁で，月刊FFEと呼ばれています。なお，最新版のFFE.Xは，CGAマガジン3号に入っています。
Tさん(枚方)：X68030でCGAシステムが展開できません。ディスクを入れてDIRすると「無効なメディアを使用しました」と表示されてしまいます。これってディスクが壊れたのでしょうか？
うさ子：そうです。
ゆたか：ちゃう！ 昨年の7月号のCGAシステムは，X68030対応してへんねん。あたりまえやがな。X68030が出る半年以上前やから。X68030対応版は，DōGAのほうに直接申し込んでください。話は変わるけど，一口大阪弁講座「この犬は，チャウチャウではないでしょうか。いいえ，チャウチャウではないと思います。いや，チャウチャウでしょう。いいえ，チャウチャウではありません」を大阪弁でいってください。
うさ子：「この犬，ちゃうちゃうちゃう。いや，ちゃうちゃうとちゃうんとちゃう。え〜，ちゃうちゃうちゃうん。ちゃうちゃう，ちゃうちゃうちゃう」
ゆたか：さすが生粋の大阪人。なお，この問題は，探偵ナイトスクープを参考にさせていただきました。
?さん：かまたさん，うさ子さん，ご結婚おめでとうございます。おふたりの名前で，「三枝の新婚さんいらっしゃい」に申し込んでおきましたよ。……ウソです。
ゆたか：うぉー，一瞬あせった〜。でも，それってどんな番組なん？
うさ子：新婚さんが出演して，いちゃいちゃするの♡
ゆたか：へ〜，いちゃいちゃ♡
うさ子：いちゃいちゃ♡
（他スタッフ一同：そやから，いちゃつく口実を与えるようなお便りはやめんかい！）

に，そのうちにやろう，やろう……と，どんどん先送りになってしまうものです。思い立ったが吉日！　まずは，10月末日に間に合うように，計画してみてください。

DōGAからのサポートについては，「執拗な催促」(？)などいくつか計画していますが，皆さんの申し込み用紙を拝見してから詳しく検討していきたいと思います。その一環として，年に1～2回ぐらい，各団体の代表者が大阪に集まって，成果の発表，意見交換，親睦を深めるための「CGA会議」なるものを開催したいと思っています。そのときは，ふるって参加してください。

CGAシステム ver.3 について

さて，皆さんが気になるところのver.3とは，どのようなものでしょうか。

定義から考えますと，コラムにありますように，CGA共通規格Ver.3に対応しているものということになりますが，それでは具体的な内容が浮かびませんね。それは当然で，まだ具体的な内容は決まっていません（おいおい）。

CGAシステムの進むべき方向性としては，より使いやすく，より表現力豊かにということになります。表現力を上げるということについては，別にver.3にしなくても，現在の各ツールの機能を上げ，新しいツールを開発することで，そこそこ対応することができます。

もっと問題となるのが，使いやすさです。現在のCGAシステムの最大の問題点は，各ツールごとにインタフェイスがばらばらなことだと思います。これは，CGAシステムver.2を開発するときに，データフォーマットの統一は行ったものの，インタフェイスの統一まではよく検討されていなかったからです。当時，どういったインタフェイスがよいか不明な点も多かったし，インターフェイスなんてほとんどないも同然のプログラムも堂々と販売されているような時代だったのです。

ですから，今回のCGAシステムの開発にあたっては，CGA共通規格だけでなく，インタフェイスの共通規格，「CGAインタフェイスガイドライン」も同時に検討していきたいと思っています。

また，ver.2の開発のころから比べると，パソコンの環境や技術も，どんどん新しいものが出てきました。マルチタスク，ウィンドウシステム，パイプライン処理……。それらをどんどん投入することが，より進んだのシステムのようにも思えますが，いたずらに新しい技術を追い求めるべきではないでしょう。現実性を考えて，アマチュアプログラマが開発に参加できるようなシステムであり，そして実用性を大切にして，真のパーソナルCGAシステムとは何かを考えたいと思います。

CGAシステムが開発されて5年間，一応の実用性を保てた秘訣として，当時のワークステーションのCGA環境を研究し，5年後にはこの環境がパソコンレベルで実現されるだろうという予測に基づいて設計したことが挙げられると思います。今回のver.3についても，現在のワークステーションのCGA環境において標準的なものは実現したいと思います。……ちょっと無理なような気もしますが，5年前はOUCCと，KMCだけで実現できたのですから，今回皆さんが協力してくだされば，必ずや実現できるはずです。ver.3の実現は，これを読んでいる皆さんにかかっています。

DōGAのスタッフになってみたら

いきなり，DōGAの紹介レポートを書けという指令が出た。これやったら学校と同じやんけと思いつつ，この春からスタッフとして参加してみた感想なんかを書いてみます。

初めまして。大阪の工業高校1年生のハイスクール飯干と申します。イベントなんかで会うかもしれないので，よろしくお願いします。

僕がX68000を買ったのは，ゲームのためでした。しかし，友達からCGAコンテストのビデオを見せてもらい，そのなかの「TORNADO」を見て，とっても感動しました。僕は，すごく車が好きです。僕もこんな作品を作りたいと思って，DōGAに参加しました。いまも，ちゃんと作品を作っています。

初めてDōGAに行く前は，きっとプロジェクトルームには機材がいっぱいあるんだろうなとか，汚いところだろうなとか想像していました。そして，いざ行ってみると……だいたい想像どおりのところでした（思ってたよりきれいだったけど）。

皆さんがいちばん心配しているのは，「DōGAって，危ないところとちゃうん」とか，「オタクがいっぱい居るんとちゃうん」とかだと思います。けど，僕の友達なんかを連れてきても，「なんや，そんな危ないところとちがうやん」といっていますし，僕もそう思います。

DōGAのスタッフになって，よかったと思うのは，CGAコンテストのビデオなんかでは見られない秘蔵作品の数々が見られることです。プロのCG集なんかもたくさんあります。それから，CGAシステムのことや，パソコンのことなど，わからないことをいろいろ教えてもらえます（裏ワザとか悪いことも……)。また，まだ経験はありませんが，いろんなイベントに連れて行ってもらうのも，楽しみにしています。

でも，そんなことより，いままではOh!Xのなかで読むだけだった人たち（かまたさんとか，MAX田口さんとか，あっ，こないだ初めてうさ子さんに会えました）といっしょに，DōGAの一員になれることが最大のメリットだと思います。

もっとも，かまたさんに嫌な仕事は押しつけられるというデメリットもありますが，かまたさんは「人生何事も経験だ」といっているので，とりあえずそういうことにしておきます。

スタッフとしての仕事は，あまり心配することはありません。プロジェクトルームにいれば，自然とわかってきます。Oh!Xでは，いつも人手が足りないように書いてありますが，暇なときは結構暇です。いきなり行って，誰にでもできるような仕事があることはほとんどありません。何度か出入りしているうちに，雑用を頼まれるようになります（まだ，大切な仕事は任せてもらったことはない)。

ですから，何も用事がないときに自分でする仕事を持っておいたほうがいいと思います。たとえば，プログラムの制作とか，作品の制作とか。そしたら，用事がないときでも有効に過ごせるし，周りの人から，いろんなことを教えてもらえます。

マリオ古本さんには，「高校からこんなところにいるとは，人生を捨ててるやろ」とかいわれたりもしますが，僕でも結構ちゃんとやっていけています。皆さんも簡単な気持ちでもいいと思いますが，ちゃんと自分の意思（何をしたいのか）を持って入ってきてください。

（ハイスクール飯干）

DōGA法人化への道　特別編

まず，プロジェクトチームDōGAは，これからもアマチュアであり，非営利団体として活動します。そういうと細かい点で誤解を招くかもしれないので補足しますと，我々はパーソナルCGAの普及を目的としており，お金儲けのために活動をしているのではない。これだけは明言しておきます。これは公約ととっていただいて結構です。できなければ，政治的責任をとります……なんのこっちゃ。

さて，いままでの詳しいいきさつは，前号までの「DōGA法人化への道」を読んでいただくとして，簡単に要約すると，

1)　DōGAの活動が広がってきた。
2)　そこで，専任スタッフが欲しい。ちゃんとした団体になりたい。
3)　ならば，公益法人になろう！
4)　しかし，残念ながらお役所の許可が得られなかった。
5)　しかたがないので一部を株式会社として法人化した。
6)　しかし，アマチュアと株式会社が混在している。
7)　さらに，ちゃんとした団体になるという目標が未達成のまま。

てな感じになります。

そこで，みんなで話し合った結果，プロジェクトチームDōGAは，今後次の目標を掲げて活動することになりました。

「DōGAは，5年後を目標に，国から公益法人としての認可を得られるように努力する」

今回，お役人さまのお許しが出なかった主な理由は，ひとつに，公益性のある活動をしているという証拠がないというのと，公益法人である以上，営利活動をしなくても活動を続けていけるという証拠がないという2点です。

前者については，たとえばOh!Xのこの連載自体が活動の証拠になってもよいはずなのですが，お役人さまのいう証拠とは，お役人さまが用意してある書式どおりに書かれた書類の束だけなのです。ですから，ちゃんと公益法人として認可を得るためには，それらを勉強して，これからの活動をその書式にして残していかなければいけません。この書類が，約5年分必要なんだそうで，ですから公益法人化の目標も5年後としているわけです。

次に，後者の営利活動をしないという証拠ですが，具体的な例としては，「社団法人○○学会では，会員が1万人おり，年会費が1万円なので，年間1億円までの予算の活動ができる」といったものです。DōGAの場合，いままでカンパによって活動資金を出していましたが，このカンパは，人数，金額ともに不定で，とても予算が立てられません。お役人さまからも，「カンパがある年はよいが，少ない年は営利活動をすることになる」と反論されることが目に見えています。

そこで出てくるのが，株式会社ドーガです。アマチュアのDōGAは，基本的に営利活動は行いません。しかし，株式会社ドーガは，積極的に営利活動を行います。そして，そこで得た収益はアマチュアのDōGAの活動資金として利用されます。こうすることで，アマチュアのDōGAは，営利活動を行わなくてよいということになるのです。

しかし，これだけではいけません。株式会社は利益を

各読者連絡事項

ビデオ発送にトラブル発生！（魔の3日事件）

順調に終了したと思われていた第5回CGAコンテストのビデオ発送作業だが，とんでもないトラブルが発生していたことが，申し込み者の指摘により判明した。ある3日間に申し込まれた方すべてに発送されていないのだ。

さっそく，緊急対策本部を設置し，申し込みの現金書留をすべてチェックしたが，その3日間のものは1つもない。現金出納帳を調べても，その3日間は申し込みが0になっている。そういえば，毎日たくさんの申し込みがあるなか，その3日間だけ申し込みがなかったのを覚えている。郵便事情で，1～2日，申し込みがなくなることは何度かあったが，おかしいなぁとはいっていた。

これは，郵便局の配送ミスだー！　と勇んで，郵便局に乗り込んだのはいいが，数時間粘って調査してみた結果，ちゃんとDōGAの受取印が押されていることが判明した。あれ～？　いったいどうなってんだぁ？

現在も調査中であるが，もうほとんど調査できることは調査済みで，完全に行き詰まってしまった。このままでは迷宮入りだ。う～，ホームズ助けておくれ～。

＊　　　＊

とりあえず，ビデオを申し込んだがまだ着いていないという方は，現金書留の控えがなくても結構ですから，申し込まれた月日だけでも明記して，至急ご連絡ください。お手数をお掛けして申し訳ございません。

指名手配

CGAマガジンの定期購読を申し込まれた「京都府舞鶴市上安1922大志寮」の方，氏名がありませんので，至急ご連絡ください。

CGAマガジン第3号のおわび

ハードディスクにインストールしたとき，アニメーションおよびデータベースをAドライブとBドライブに展開できないというバグがありました。

ディスク1の ¥BAT¥HD.K, ¥BAT¥ALL.K, ¥BAT¥DB_HD.Kを図1のように修正してください。

HD.K　：33行目
DB_HD.K：33行目
ALL.K　：58行目

図1

（誤）
```
if (substring(work_dir, 2, 2) != ':' || work_drv == "A" || work_drv
== "a" || work_drv == "B" || work_drv == "b" || checkdrive(work_drv)
 == 0)
```

（正）
```
if (substring(work_dir, 2, 2) != ':' || checkdrive(work_drv) == 0)
```

追求するための団体なのに，その利益をすべて寄付するというのは，単なる脱税行為とみなされてしまうのです。そこで，アマチュアと会社間で，次のように契約を交わします。

・アマチュアのDōGAは，株式会社ドーガに対して，DōGA CGAシステムの独占的商業利用権を与える。

・株式会社ドーガは，その代償として，アマチュアのDōGAの活動資金を出す。

会社がお金を出すのは，あくまでも使用料なので，これは正当な出費とみなされます。これで，すべてのつじつまが合ったわけです。

えっ？　なんか騙されたような気がするって？　実は私もそういう気がしないでもないですが，税理士さんとも相談しましたが，この理屈はちゃんと通るそうで，実際そんな例も多々あるそうです。お役人さまは，つじつまさえ合えば文句はいわないのかもしれませんね。

株式会社ドーガの位置づけ

わかりやすくまとめると，アマチュアのDōGAは，これまでどおりパーソナルCGAのために，非営利の慈善事業を行います。そして，これを資金的に援助するために，株式会社ドーガは，営利活動を展開していきます。つまり，アマチュアに対しての活動はアマチュアのDōGAが，営利組織に対する活動は株式会社ドーガが行うわけです。

スタッフや皆さんが開発したいろんなプログラム，データは，その使用権，配布権などをアマチュアのDōGAに譲与してもらい，アマチュアのDōGAで一括管理します(当然ながら，各種の権利は本人がもったままです)。基本的にDōGA CGAシステムは，営利目的の使用を禁じ，あくまでもアマチュアに使ってもらいます。そして，唯一営利目的の使用を認める団体として，株式会社ドーガがあるわけです。ですから，何らかの理由で，CGAシステムを使った営利活動を行う場合，事前に株式会社ドーガの許可を得，その収益の一部を会社に支払わなければいけません。

株式会社の営利活動の内容としては，CGAシステムによる映像制作業務を行う予定です。アマチュアのDōGAがやっていたようなことで営利活動するのはできるだけ避けたいと思っています（アマチュアのDōGAといっしょくたになるから)。ゲームのソフトハウス向けのデモ用の映像や，地方テレビやCATV向けの番組タイトル，一般企業向けのビジネスプレゼンテーション用の資料映像などを制作するつもりです。

しかし，ここで根本的な疑問がひとつ浮かびます。「株式会社ドーガは，DōGAに資金提供する以前の問題として，ちゃんと収益上がるんかいな？」

そうなんですよ。これらの計画は，会社が十分儲かるという計画に基づいていますが，そんな保証，どこにもないんですね。困ったことに。

確かに，ほかのCGプロダクションより有利な点もあります。しかし，会社の運営資金だけでなく，アマチュアの資金まで捻出するとなると……。ということで，株式会社ドーガへの支援も，間接的（直接的？）にアマチュアのDōGAへの支援となると思って，ご協力ください。名づけて「Oh!Xの読者10万人営業マン計画」。キャッチフレーズは，「CGA制作のご依頼は，安い，早い，おもろいのドーガへ！」

おわりに

「DōGAに参加するったって，うちのクラブの現状を考えると，興味あるヤツはいるかもしれんけど，そこまでやる気のあるヤツはいるかなぁ」という方も多いでしょう。しかし，そこは考えようです。先に，DōGAへの参加を決めてしまいます。そうすれば，あなたのクラブは立派なDōGAグループの一員となります。

そうして，CGAシステムやCGAコンテストの作品などを有効に利用し，来年の新入部員にPRします。そして，このクラブはこんな活動に参加していて，今年はこんなことをする計画だと教えるわけです。そうすれば，必ず最初からやる気のあるヤツが集まるはずです。あとは，あなたを中心に，1回生を指導していけば，立派な活動ができるでしょう。やる気がないからできないのではなく，これをきっかけに，やる気のあるクラブにしていけばよいのです。

*　　*　　*

ところで，昨年の「第1回全日本X68000芸術祭」のゲーム部門応募作品の優秀作品5点がまとめられて「The World of X68000」として発売されました。そのマニュアルのなかで，「ああっ！お姫さまっ！」の作者の高倉氏がその原稿料，印税をDōGAにカンパされると書いてあったのですが……？　当方としては，高倉氏からそのような多大なカンパをいただく理由がないので，戸惑っております。また，第4回のCGAコンテストのビデオを家宝にしていらっしゃるそうですが，せめて第5回のビデオぐらいお送りいたしますので，これをご覧になりましたら，DōGAまでご連絡ください。

*　　*　　*

さて来月は，ソフトバンクから創刊される某雑誌に出張するためお休みをいただきます。そして，再来月は，CGAシステムのマニュアルのなかで休講とされていた，CGA大学の「EPA2」の補講を行います。このEPA2.Xは，世間一般からペイントソフトと勘違いされているようですが，いろいろ面白い使い方ができるんですよ。

楽しみにしていてください。

目指せジョイスティックの星〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜〜(2)

理想を手に入れるための道

Ishibumi Akira
伊澁見 あきら

壊れたジョイスティックを究極のスティックに改造しようというこの連載。今回は基本的な構成とそれを実現するための電子回路を設計します。でもいちばん大変だったのは穴開けだったりして……。

いきなりで恐縮ですが，忘れないうちに書いておきましょう。今回からは「ローテク」ではありません。前回(すごい昔のような気がしますが，たった3カ月前の7月号です)では，特集に合わせることを考え，線を切ったりつないだり程度のレベルに抑えていたのですが，究極のジョイスティックを作るということになると，ローテクのままで済ますというわけにはいきません。

そこで，あらゆる制約を取り外し，X680x0用の究極ジョイスティックの名に恥じない代物をでっちあげようという考えに至りました。では覚悟を決めて，究極への道への一歩を踏み出すことにしましょう。

▶その前にちょっと復習◀

前回，最後の工作のところの説明が少ないという指摘がありました。なかでもケーブルやコネクタの説明不足という点には，書いた本人も反省しているので，この場を借りて，フォローすることにします。

まずは，X680x0のジョイスティック端子そのものについてです。本体の取扱説明書を参照すると，一応それについてのピン配置などが載っています。しかし，これが曲者で「8255のPA0端子」などといった，意味不明な記述がされていて，何番ピンがなんの入力に使われているのか明記されていません。これでは困ると思ったので，あえて図1にまとめてみました。

この場合，X680x0側のジョイスティック入力端子に正面から向かって，左上から上下左右の順になります。前回もわざわざ書きましたが，8番ピンはコントロール出力であって，0Vではありません。入力装置のジョイスティックには，本来無縁のものということになります。君子危うきに近寄らず，余計な欲は出さずに，この出力には手を触れないことにしましょう。

また，すでにコネクタ部がつながっているケーブルや，延長ケーブルのジャック側を切り落として使うことにした場合についての注意も書いておきましょう。

コネクタ部が樹脂で固められているタイプを使えば，すでに線が半分つなげてありますし，ツメがX680x0側にぶつかる心配もしなくて済むので，一石二鳥のような気がします。しかし，実際には同軸ケーブルの内部の線の色とコネクタのピン番号の対応は，かなりマチマチです。抵抗のカラーコードと似ていたり，まったく別物だったりします。原則的に，作業の前に毎回テスターなどを使い，ピン番号とコードの色の対応をチェックすることを，お勧めします。

経験的に，同じ店でいつも売っているような場合はモノが同じだと思われますし，内部も多分同じだと想像はできるのですが，さすがに100%断定することはできません。特に延長ケーブルの場合，コネクタが同じように見えても，作っているところが違っているおそれがあるので，用心したほうがよいでしょう。

▶目標と煙は高いところに昇る◀

では，本来の目的に戻り，X680x0の究極のジョイスティックを具体的に考えてみましょう。

まず最初に押さえなければいけないのは，変更できない固定された仕様の部分です。つまりX680x0側に出力するところの仕様です。いくら，ジョイスティック側でレバーやボタンを増やしたとしても，その信号を受け取る側に入力端子が用意されていなければ，余分な入力デバイスは，ただの飾りになってしまいます。

つまりこれは，方向用のレバー1個分と，ボタン2個分の信号以外の情報を，ジョイスティック側から送っても無駄ということを示しています。これが，機能を設計するうえでの前提条件になるわけです。

では，実際にあったら便利だと思われるような機能を考えてみましょう。ほとんどの機能が，すでに完成している自分愛用のジョイスティックで採用したものですが，ここで，それらがどういった経緯で生まれたのかを振り返ることで，ジョイスティックのチューンアップの考え方を説明できると思います。

まず最初に考えつくのは，連射装置でしょう。ボタンを押したままで，高速自動の連続攻撃が可能になります。X680x0でのシューティングゲームにおいては，より多くのプレイヤーに門戸を開くという意味で，初心者からマニアに至るまで非常に要求や価値の高いものだといえます。もちろんどちらのボタンも，独立してそのオン・オフができるのは，当たり前のことでしょう。

ほかには，2つのボタンの左右反転があります。これは最近ではほとんどソフトウェア側でサポートされ，わざわざジョイス

図1　ジョイスティック端子

ジョイスティック2

端子No.	信号名	I/O	備　考
1	IOB0	In	8255のPB0端子(上)
2	IOB1	〃	〃　PB1 〃(下)
3	IOB2	〃	〃　PB2 〃(左)
4	IOB3	〃	〃　PB3 〃(右)
5	Vcc1	Out	+5V
6	IOB5	In	8255のPB5端子(トリガーA)
7	IOB6	In	〃　PB6 〃(トリガーB)
8	IOC5	Out	〃　PC5 〃(コントロール)
9	GND	−	グランド

ティック側で対応する必要はないのではないかという意見もあるでしょう。

しかし，一部の変更のきかないソフトをプレイするときも，設定モードでわざわざ変更しなければならないときも，手元だけで同じ作業で済んでしまえば，面倒がなくてお手軽です。

しかも，ソフト側で変更できたとしても，まずそれらの設定が保存されることは珍しいので，結局何度も設定することになります。考えようによってはこれも，必須の機能といえるでしょう。

そして最近，シャープ純正のパッドやスティックではサポートしていないのに，次第に定着しつつあるSELECT，STARTの２つのボタンも無視することはできません。

これらは，原理的には非常に単純なもので，ジョイスティックの同時には入らないハズの2つの方向を同時に入れてしまうというものです。SELECTを押すと左右が同時にオンになり，STARTは上下が同時にオンになるという仕組みです。

対応ソフトは少ないですが，手元でちょっとポーズとかできるとやはり便利です。本来はFM TOWNSの純正パッドの機能ですが，今回改造中のスティックはもともとファミコンのジョイスティックなので，都合よくボタンがついていることもあり，これも採用することにしました。

▶究極はより高いところに◀

このようにいくつか汎用的な機能を考えたところで，これらは市販品で見ることのできる機能ばかりです。究極と銘打ったからには，さらに強力な機能を装備する必要があるといえます。ではそのプラスアルファを見つけるには，どうすればいいのでしょうか？　そのヒントは，具体的なゲームをプレイするスタイルをシミュレートすることで，はっきりした姿となって見えてくるはずです。

まずは連射装置について，より高度な連射の姿を考えてみましょう。ちょっと凝っている連射パッドやスティックで可能な，２つのボタンの速度を変えて連射するプレイが要求されるようなゲームは，私の知っている限りではひとつもありません。

むしろドラゴンスピリットで，２つのボタンを押したままで，空中攻撃と地上攻撃を交互に連射してくれると，空中と地上を同時に攻撃できて，かなり便利です。これは２つのボタンごとに連射速度を変えられたりするタイプでは，実現不可能です。連射がボタンごとに独立しているのが，かえって利用価値を下げているのです。

すると連射の速度は１種類で，２つのボタンを一緒に連射するか(同期式)，交互に連射するか(反転式)というように連射の方式が選択できるほうが，利用価値があると考えられます。

さらに連射について考えると，連射と押しっぱなしを使い分ける必要のあるソフトでは，頻繁に連射のオン・オフを切り替えながらプレイすることになります。これは非常に面倒で，迅速性やプレイのリズム感を損ねます。グラディウスシリーズや今度発売されるコットンなんかが，典型的な例に挙げられるでしょう。

同じショットボタンを２つ用意して，片方が連射で片方は通常のボタンというような使い分けができればよいのですが，これを２つボタンのままでやるには，どう考えてもボタンが足りません。これの解決には，ジョイスティックのケースに穴を開けて，３つ目のボタンをつける以外の方法はありません。

そして，ボタンの左右逆転についても，３ボタン方式を前提にするならば，単純に左右を逆転させるというわけにはいきません。かといって，いくつかのボタンの割り当てパターンを固定してしまうというのは，あまりエレガントではありません。連射・溜め撃ち併用のため以外にも，ファイナルファイトや餓狼伝説で，両方のボタンを押して行う必殺技や，投げ技を３つ目のボタンで使えると，非常にプレイがスムーズになることは，想像に難くありません。

一歩進んで，３つ目のボタンの機能だけをいくつかのものから選ぶのではなく，ボタン全部を平等に扱い，自由な機能を割り当てることで，高度なボタンの自由度が確立することになります。

▶究極を手に入れるために◀

このようなことを踏まえると，究極のジョイスティック実現のために，装備するべき機能が見えてきます。

まず第１に，ボタンを３つにします。なおかつそのボタンは，どれもX680x0における２つのボタン入力のいずれにも割り振ることを可能にします。そして，ボタンごとに連射機能のオン・オフが設定でき，ボタン同士の連射信号の同期・非同期も選択できるというものになります。あと，SELECTとSTARTボタンも，一応サポートしておきます。

これが，試行錯誤により生み出された，X680x0用究極のジョイスティックの付加機能です。では，この妙ちくりんな要求を満たすためにはどうすればいいのでしょうか。いくら理想の機能を並べ立てても，具体化できなければ，絵に描いた餅に終わってしまいます。その解決法を探っていくことにしましょう。

まずは３つ目のボタンを取り付けるため

図2　回路図

部品表

タイマーIC　555	×1
74LS04	×1
74LS11	×1
R_1：1kΩ	×1
R_2：10kΩ	×1
VR_1：47kΩ	×1
C_1：1μF	×1
中点OFF型3Pスイッチ	×3
2Pディップスイッチ	×3

に，穴を開けなければいけません。しかし今回のような，比較的小さいボタンを使用しているタイプでも，穴の直径は20mmを超えてしまい(今回は24mm)，ちょっとした日曜大工の工具程度でキレイな穴を開けることは至難の技というしかありません。

適当に太めのドリルで穴を開け，ヤスリで整えるとか，ハンドニプラーで切り開くという方法では，穴の縁が歪んでしまったりするのはもちろん，そもそも綺麗な丸い穴を開けること自体に難があります。

そこで工具屋で探してみると，挟んでネジで打ち抜くように空けるパンチと，ドリルの先につける鉄板の穴開け用の刃の2つが見つかりました。今回は，開ける穴の大きさの都合で，ぴったりのサイズが穴開け用ドリル刃のほうにしかなかったので，こちらを採用しました。鉄板用で直径20～30mm前後ならば，2,000円しないような価格ですので，それほど高い工具というわけではありません。これで穴開けのほうは準備ができました。

▶肝心の回路の設計◀

次はボタンの連射化と，配置の自由化ですが，こういった，ややこしい機能を実現するのは，電子回路と決まっています。そもそもスイッチだけとはいえ，ジョイスティックも立派な電子回路です。またX680x0側の入力が，TTLというデジタル回路用ICの信号規格を採用しているので，もちろんそれに準拠して回路を設計することにしました。できあがった回路図は，図2を参照してください。念のために使用したICのピン配置も書いてあります。

この回路は，実際にはかなりの試行錯誤からたどりついたもので，詳しく説明したいのですが，あまりハードウェアに詳しくない人にクドクドと説明しても退屈なだけだと思います。しかしあえて，ひととおりの説明をしておきたいと思います。

まず連射回路の発振源にしているのは，初心者の電子工作などでお馴染みの，タイマーICの555です。市販品の連射装置ではこれとは違ったデジタル回路の発振回路を使っているものが多いですが，要求される発振周波数が数十Hzと低いことと，その安定性から，この方式を採用しました。

ちなみに回路は，ICを買ったときについてきたデータシートの推奨回路をそのまま使っています。発振周波数範囲は12～72Hz程度です。一般にゲームソフトの場合，垂直同期に合せてプログラムが入力をチェックするので，1/60秒ごとに，交互にオン・オフするのが連射の最高速度になります。そのため，この回路では30Hz前後で微調整がきくように部品の数値を選んでいます。

また，連射の切り替えは，TTL規格のICの入力端子を開放したときに，5Vと等価で扱われるというコスい手口を使用しています。このため74HCシリーズのような，CMOSタイプのICを使うと，この回路は動きません。

つまり，真ん中でオフになる3Pスイッチの信号を，いったんデジタル信号的に反転し，ボタンを通してつないでいます。配線が面倒ですがこのほうが電子部品の数は少なくなります。スイッチには，連射信号が反転した信号と一緒に供給されているので，各ボタンに対応したスイッチを，同じ側につなぐか，逆側につなぐかで，連射の同期・非同期を決めることができます。

最後の出力部では，各ボタンからの信号のOR回路を用意しておいて，あとはそれぞれのボタンと，ディップスイッチでつなぐかどうかを選択する仕組みになっています。

予想はしていましたが，このようにスイッチがかなり増えたために，ジョイスティックの底に穴を開け，基板上に取り付けた連射速度調整のボリュームと，ボタン配列のディップスイッチを，その穴から操作するようにせざるをえなくなってしまいました。しかし，これで回路のほうも準備は完了しました。

▶究極の道は遠く◀

とりあえず今回はここまでです。次回はこれらの実際の組み立てと，メンテナンスなどの話も書こうと思っています。最初は連射とボタンの左右逆転をつけて終わりにする予定だったのですが，いつのまにやら話が大きくなって，ここまできてしまいました。しかし，計画倒れの域は脱していますので，あとはもうひたすら作るだけです。はたして無事完成するのかどうかについては，次回のお楽しみということにしておきましょう。

このように穴を開けた

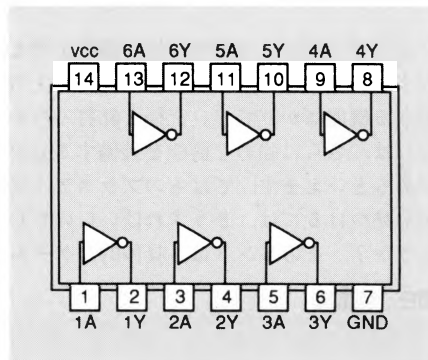
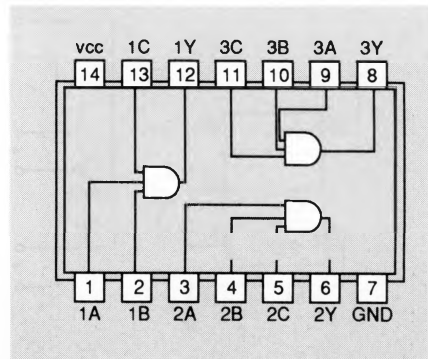
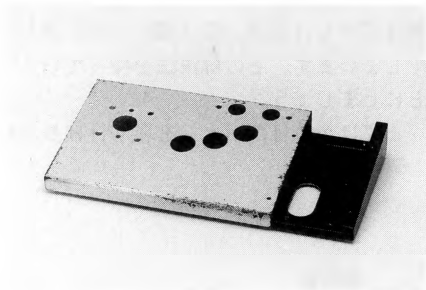
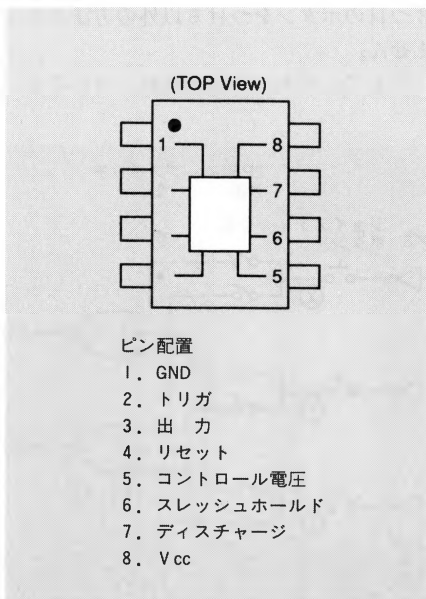
今回使われた工具たち

図3　555端子表

図4　LS04端子表

図5　LS11端子表

CREATIVE COMPUTER MUSIC

Creative Computer Music入門(25)

和声の発生

ひき続き，音楽理論が形成されるもとになった原理について解説します。今回のテーマである和声も全音階や半音階と同様に倍音列から発生しています。以前に解説した和声進行などの理論の裏付けになりますので，そのとき理解できなかった人ももう一度考えてみてください。

Taki Yasushi 瀧　康史

自然和音の発生

前回は，倍音列から全音階，半音階が形成された原理を紹介しました。今回はその続きで，和声の発生の原理を考えてみましょう。そこでまず本題に入る前に，前回の内容についてちょっと復習します。これは今回説明することに密接に関わる部分があるからです。

図1は先月号でも掲載したものですが，Cを基音とした倍音列です。一般的に2つの音は，その振動数の比が単純であるほど大きく振動します。最も単純な振動数比は1：2で，これはオクターブつまり完全8度を意味しています。この場合の2つの音はあまりにも重なり合ってしまうので，実際には音の高さが違うにもかかわらず，通常私たちは「同じ音」として扱います。

振動数比が1：3になると，オクターブ違いの音に比べて，音の混じりが若干悪くなります。音の響きという点ではよく協和する音ですが，同じ音に聴こえてしまうというほどではなく，私たちの耳にもまったく違う音として把握することができます。つまり，振動数比1：2の場合は同一の音の重なりであったのに対して，1：3では確実に違う音同士のハーモニーになるといえます。よって，振動数比1：3というのが，異なる音同士の音の関係のなかでは最も基本的なものを表していることになるのです。

そして，この振動数比が1：3つまり完全5度の関係をもとに基本的な音階が生まれたのです。

以上が，前回解説したことのおおまかなまとめです。このあたりの詳しい説明については先月号を読んでもらいましょう。

さて，自然和音の発生ですが，これはこの音階の発生と密接な関係があります。

人間がいちばん初めに得たハーモニーとは，完全8度です。これは，男女が同じ旋律を斉唱するときにすでに表れます。たとえば，図2の2音は完全8度の関係にあります。しかし，この完全8度というのはオクターブ違うだけの「同じ音」なので，ハーモニーと呼ぶにはいささか不適当です。そして，音の厚みが出てきません。

そこで，倍音列のなかから完全8度の次に美しく協和する完全5度（振動数比1：3）をもってきて和音を作ります(図3)。この2音は明らかに違う音でありながらよく協和するため，和音として扱うことができます。この完全5度の発見により，人間は初めて厳密な意味での和音を得ることができたのです。

しかし，この2和音だけではあまりに協和して融け合うので，それぞれの音の独立性は弱められてしまいます。したがって，和声そのものを考えるといささか空虚さが感じられます。そこで，3倍音である完全5度に次いで単純な振動数比である5倍音の長3度（別名自然3度ともいう）の音を加えて，3和音を構成することにしました(図4)。これによって，かなり充実した響きを得ることができます。

これが最も基本的な3和音で，長3和音または自然3和音といいます。

次に，この自然3和音を，全音階の各音の上に配置することを考えてみます。しかし，この時点ではまだ半音階は生まれていませんので，全音階のなかのすべての音

図1　Cを基音とした倍音列

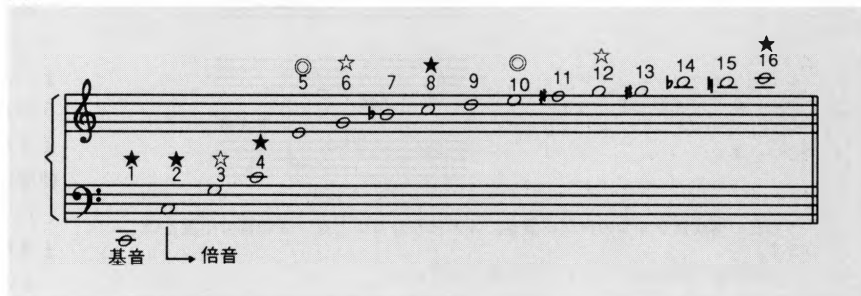



図2　完全8度の2音による和音

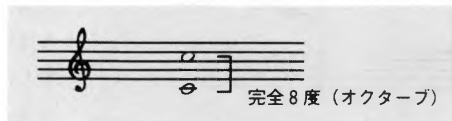



図3　完全5度の2音による和音

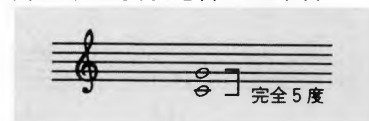



図4　長3和音（自然3和音）の形成

の上でこの自然3和音を構成するのは原理的に不可能です。そこで，類推的に各音の上に3和音を作ってみます。すると，必然的に短3和音と減3和音が生まれてくることがわかるでしょう。これは図5のようになり，これらをダイアトニックトライアドコードといいます。ダイアトニックは音階を表し，トライアドコードとは3和音を表します。

しばらくのあいだ，これらの3和音は音楽を構成するうえで十分な和音と思われてきました。つまり，古典音楽においては，2和音とか4和音は存在せず，仮に2和音があったにしても，それは3和音の省略形として扱われました。また，4和音は3和音＋非和声音と考えられてきたのです。

§ 4声体（混成4部）

音楽が人間の声から生まれたことは前にいいました。

人間の声を，女声と男声の2つに分け，さらにこの2つのなかでそれぞれ高い声と低い声とに分けたとき，これらは全部で4つに分けることができます。女声と男声のあいだにはオクターブの差があり，同性の声のなかの高いほうと低いほうのあいだには5～4度の音度差があります。これらを上から順にソプラノ，アルト，テナー，バスといいます。これらを混成4部といいます（コラム1）。

中世には，多声音楽の発展とともに，5声以上の混成が組まれたりしましたが，その基本はあくまで4声でした。これは，人間の声の性質からきているといってよいでしょう。

さて，倍音列は低音域において広く，高音域では狭くなっています。したがって，低音部分に音を密集すると音はこもりがちになり，高音部の和声を省略すると音が散漫になりがちです。このことは，作曲をするときの4声体の配置にも密接に関係します。つまり，以前に私たちが4声体を学習したときに述べたように，ソプラノとアルト間はオクターブ以内，アルトとテナー間もオクターブ以内，テナーとバス間は12度以内に配置するべきなのです。

また，倍音列を下から順に読んだとき，最もひんぱんに現れるのは基音で(倍音2，4，……は同じ音)，次いで，倍音3，5です。この倍音列のなかでの出現頻度は，曲のなかで和音の反復をする際の音の出現頻度とほぼ同じになります。図1の中で★のついている音が基音で，☆が倍音3，◎が倍音5です。

旋律進行

旋律とは，いいかえるとメロディラインのことです。メロディラインすなわち旋律進行の善し悪しを，一意的に歌いやすい旋律音程と定義します。これはつまり，私たちの音感覚で把握しやすい音程を，よい旋律音程と定義したということです。

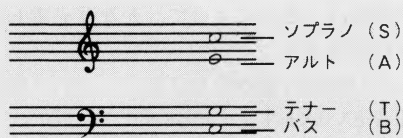
この定義から2つの定理が生まれます。

ひとつは，近い音程の移動ほど把握しやすく，歌いやすいという事実です。例を挙げれば，いきなりCからAまで飛ぶような旋律進行よりも，CからDに移動したほうがDの音を把握しやすいということでしょうか。このような2度を連続度といいます。旋律進行において，連続度は最良の旋律音程といえます。以前に4声体を学習したときに，できるだけ近めな音に移動するというのはまさにこのことなのです。

もうひとつは，2音の調和です。自然3和音のなかに含まれる完全5度と，長3もしくは短3度の転回音程を求めると完全4度と短もしくは長6度になります。これらは完全5度，長3度の次に私たちの音感覚で把握しやすく，そしてよく協和するので，オクターブ（すなわち完全8度）とともに，協和音程といわれます。連続度（2度）以外に旋律音程と認められる音ができました。それ

コラム1

4声体のうち，4声部は次のように命名されています。

上から順にソプラノ，アルト，テナー，バスです。一般に，ソプラノは女性の高い声，アルトは女性の低い声，テナーは男性の高い声，バスは男性の低い声です。

4声体のうち，ソプラノとバスを外声，アルトとテナーを内声といいます。

上方の3声（ソプラノ，アルト，テナー）を，上3声，下方の3声(アルト，テナー，バス)を，下3声といいます。

3和音の構成音を4つの声部に置き，4声体を作ることを，3和音の配置といいます。

4つの声部の音域はだいたい次のとおりです。

図5　ダイアトニックトライアドコード

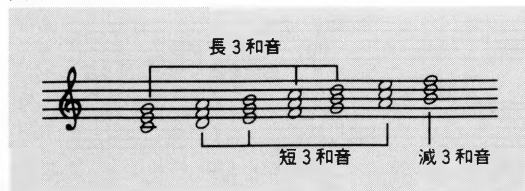



図6　連続度（2度）以外の旋律音程

を示したのが図6です。

その結果，4声体の旋律進行で，避けなければならないのはこれ以外の音程，すなわち長・短7度と，増音程，複音程（9度より広い音程）となります。

§ 和声進行

旋律進行が単一の音の進行の規則なのに対して，和声進行は重音の進行の規則です。和声進行についても1992年11月号で触れましたが，注1を参考にしてください。

原則として和声進行は，各声部の独立に気を配らなければなりません。これはつまり，2声部を考えればわかります。2声部が常に2声部であるためには，双方の声部が常に独立していなくてはなりません。それぞれの声部が独立していない場合，これはすでに和声が崩れているといわねばなりません。

次に挙げるものは和声進行を破壊するため，禁じられている進行です（図7）。

1）　連続1度進行

2つの声部がまったく同じように同一の音に進行する場合，それは事実上，2つの音ではなくひとつの音になってしまう。よって，この進行は各声部による独立性がなく，和声進行を破壊する。

2）　連続8度進行

オクターブ離れた場合も1)と同様に，同じ音になるので各声部による独立性がない。

3）　連続5度進行

2声部が5度で連続進行した場合も，1）や2）の場合ほどの結合力はないにしろ，独立しているというには不十分といえる。よってこれも不可。

これらの禁じられた和声進行は，1992年11月号で4声体を学習したときにも解説したものです。もしもお持ちなら，もう一度読んでみてください。

さて，本質的な旋律音程は先ほど述べたとおり，連続度進行です。したがって，特にこれといった理由がないかぎり，音は最も近い音へ進行するのが自然な印象を与えます。

連続進行に対するのは跳躍進行です。しかし，この跳躍進行は，中間が省略された連続進行ともいえます。したがって同様に考えると，直行1度，直行8度，直行5度は，潜在的に連続1，8，5度を含んでいるので，結果として，これらも独立性を失った進行になりがちです（図8）。

ただし，それでもやはり跳躍進行は連続進行とは異なりますので，それなりの条件のもとでは，許される場合があります。

1）　直行1度は和声的に軽くなるため，常に不可。

2）　直行8度，直行5度は，外声の場合のみ禁じられ，片方が内声，ないしは両方が内声であった場合は許されます。しかし，上声が連続度進行する場合，声部誘導が行われたとみることができ，これは黙認できます。

したがって，図9のような進行は許されます。

§ まとめ

今月は和声の発生についてまとめてみました。いままでの説明を補うために，細かいことまで書きましたが，いかがでしょうか。ほとんどが，前回の発展，そして1992年の11月号に説明したことの裏付けになっています。

前回，4声体について話したときは，原理には触れなかったので，納得できない部分もあったのではないでしょうか。とりあえず，音楽理論というのは，耳がそれを認めるか否かということが根底にあり，それに対する定理だといえます。なかなか認められない人もいるかもしれませんが，情報整理がうまくいった段階で納得できるのではないでしょうか。

来月は和声の機能について話したいと思います。

ではまた。

注1　和声進行

和声進行は，次の3つに分類されます。

1）　直行進行
2声部が同じ方向へ進む進行

2）　反行進行
2声部が違う方向へ進む進行

3）　斜行進行
1声部のみ動き，片方は動かない進行

また，連続進行とは，完全1，5，8度の間隔で2声部が進行することです。

図7　禁じられている進行

図8

図9　許される跳躍進行

バックアップ思い立つとクラッシュ

マーフィーの法則

『マーフィーの法則』という本を読んでみましょう。面白いですよ。これは，Oh!X先月号の新刊書案内のコラムでも紹介されていましたので，そちらのほうも見てください。ちなみに，そのコラムには「この本を頭から通読しようと思ってはいけない」と書いてあるのですが，そこを読んだときにはもう頭からの通読が終わってしまったあとでした。間違ったことをしたのでしょうか？

この本には，マーフィーの法則，あるいは，そのさまざまなバリエーションといえる法則がたくさん(たぶん1000以上)載っているのです。法則といっても，別に物理的あるいは科学的な法則ではなく，まあ，「世の中こんなものだ」的なノリの言い回し，ことわざ的なものです。

第1番めの法則(それほど面白くはないのですが)，これはマーフィーの法則そのものです。それは，

マーフィーの法則
失敗する可能性のあるものは失敗する。

というものです。これはまあ文字どおりなのですが，悪い方向へ物事は進んでいってしまうとか，ひとことでいえばアンラッキーということを表しています。

僕が気にいっているものをひとつだけ紹介しておきましょう(これは，あまりマーフィーの法則っぽくもないのですが)。

ジョーンズの法則
窮地に笑える者は，責任をなすりつける人をすでに思いついている。

最悪ケースを想定する意味

僕にいわせれば，マーフィーの法則は，「ワーストケース(最悪の事態)を覚悟しましょう」という道徳的教訓そのものです。何をやるときでも，心の片隅にいちばん悪い状態を想定できていれば，何が起ころうと，どんな結果になろうと，まったく動じないで平然と冷静に物事に対処できるというものです。

もちろん，最悪の場合を想定するとそちらに引き込まれていってしまうようなタイプの人にとっては，かえって逆効果となりますが。ワーストケースを知るということは，それほど意識的ではないかもしれませんが，誰もが頭のなかで行っていることでしょう。

ワーストケースを強く意識しなくてはならない分野もあります。たとえば，計算機で何か計算させるときにどれくらい時間がかかるかというような場合です。

たとえば，データをあいうえお順にソーティング(整列)させるときに，どういうアルゴリズムを使うかという選択が必要となります。そのときに，計算するために必要な時間の平均をその判断材料にすることもありますが，どんな場合でも最悪この時間かかるという最悪ケースの時間で比較する場合もあるわけです。そもそも，あるアルゴリズムの平均的な計算時間を求めるよりは，最悪時間を求めるほうが簡単な場合が通常ですし。

捏造版マーフィーコンピュータ学

この本は読者に対して十分な共感を引き起こし，しかも書かれている法則群がほぼ同一のテーマをもっているので，自分でもいくつでも書けそうだという気にさせてくれます。というわけで，私もちょこちょこっと作ってみましたので，ご覧にいれましょう。

ひとつふたつは気にいっていただけることを期待しています。テーマ別に分けていきます。なお，すべて自分で考えたつもりですが，当然のことながら本書の影響を多かれ少なかれ受けていますので，そのへんは了解してください。

[プログラミング]

まずはプログラミングに関することから始めましょう。プログラミングというと，どうしても不敵な存在であるバグ抜きには語れません。

バグ永久繁栄則
バグが潜んでいれば必ず悪さをする。バグのないプログラムはこの世に存在しない。

プログラミングの手間削減のための金言
プログラムを1つ書けば計算の答えがわかる。2つ書くとどちらの答えが正しいかわからない。

プログラマの定義
プログラマのエキスパートとは小さなバグをひとつも作らずに決定的な欠陥の含まれたプログラムを作る人のことである。
(注：この定義は，バグ永久繁栄則より間接的に導出可能である)

バグの悪質度に関する仮説
すんなりコンパイラを通ったときほど悪質なバグが潜んでいる。

プログラマ責任転嫁法則
バグはプログラマの最後の避難所である。
(注：これは次の法則と意味的には等価である)

窮地のプログラマに関する法則
せっぱ詰まっているのに平然としているプログラマは，すでに原因をシステムのバグにすることに決めている。

[計算機と人類]

計算機という存在と人間との関わりのことに関していくつか挙げてみましょう。

パニック時における最悪選択仮説
システムがパニックになると人は最悪の選択をする。システムの自動化が進めば進むほどその確率は高くなる。

新しいテクノロジーに関する近道原理
計算機を使いこなすいちばんよい方法は，使いこなしている友人をもつことである。

コンピュータと人間の幸福に関する金言
コンピュータとともにいるときがいちばん幸せな人は不幸せな人である。

計算機に関する人類の3分類
計算機に対する考えで，人は3分類できる。計算機は難しいと思う人，計算機は簡単だと思う人，計算機は簡単だと思おうとしている人だ。

[計算機ユーザー]

計算機とのつきあいのなかから生まれてくるものをいくつか。

viとemacsへの評価における通説
emacsをよく知らない人ほどそれをほめる。viをよく知らない人ほどそれをけなす。
(注：自分がviに執着しているからこんなことをいっているのでしょう)

マニュアルの読み方に関する法則
マニュアルのうまい読み方は，何ができないかを素早く読み取ることである。
(注：広告ではないのですから，何ができて何ができないと書いてほしいものです)

メイルとプライバシーに関する逆説
大事なメイルを読んでいるときほど，いちばん読まれたくない人が後ろに立っている。

パスワード変更に関する法則
パスワードを変えると忘れる。メモに書くとそのメモをなくす。

[計算機購入]
計算機を買うのに失敗するか成功するかということは一大事です。

パソコン購入に関する法則
パソコンの新製品をよく考えないで買うと失敗する。よく考えていると次の新製品が出る。

MIPS値に関する通説
MIPS値はそれよりは絶対速くはないという目安である。
(注：これは計算機アーキテクチャの大先生もいっています)

値下げタイミングに関する法則
買った次の週にパソコンの値下げが発表される。
(注：マックの場合は特にそうです。でもまあいいことではあるのでしょう)

[計算機一般]
計算機関係のいろいろな方面をここに集めます。

パソコン不要説
投資額に見合った働きをしているパソコンは平均すると97%である。

ユーザーインタフェイス到達度と使いやすさの反比例説
ユーザーインタフェイスに凝れば凝るほど使いにくくなる。

計算機における汎用＝無用仮説
コンピュータが汎用であるということは何にも使えないということである。

illustration : Haruhisa Yamada

コマンド使用率に関する逆説
使わないと思ってコマンドをテープに落としたとたんに必要となる。

バックアップに関する法則
そろそろバックアップしようと思った半日後にクラッシュする。
(注：いまなお計算機ユーザーの最大の恐怖はハードディスクの中身がパーになってしまうことでしょう)

計算機のコストに関する逆説
計算機が故障する確率はその値段に比例する。
(注：まあ別に逆説でも何でもないような気もしますが)

某社の方針
保証期限を過ぎると壊れます。
(注：別にタイマー内蔵で故障させているわけではないのでしょうが……)

責任分散法則
なるべく多くの人でソフトやハードを作れ。責任を押しつけることができる。

[大学，研究]
最後は，大学や研究に関することです。あまり，個人的な体験を物語っているわけではありませんので，ご勘弁を。

研究引き継ぎ不可能原理
次に来る学生に引き継がせたいようなよい研究をする学生ほど引き継ぎに熱意を示さない。

大学における学識伝達不可能原理
先生のいうことがよく理解できる学生はもともと先生のいうことを聞こうとはしない。

業績不確定仮説
論文を書いても別の人が書いたことになっている。
(注：意味不明)

データ量と信頼性に関する法則
あまりに十分なデータを示して自分の研究の優位性を主張する人ほど怪しい。

造語数と独創性の反比例
聞き慣れない造語をたくさん使う人ほど彼の研究の独創性は小さい。

後追い法則
誰もやっていない研究だと思って始めるとすでにやっている人がいる。誰かがやっていると思ってやらないと誰かがあとで一番乗りする。
(注1：研究に限らず，今回のこの企画にしても，こんなのを何十もひねり出しているのは一番乗りに違いないと思ってやっているのですが，どうせ誰かがすでに作っているのです)
(注2：ここまで書いてほっと一息ついて本誌先月号のFILESのページを読んだら，さっそくありました。ASCII誌上で特集が組まれているそうな)

参考文献
アーサー・ブロック著マーフィーの法則，アスキー出版，1993.

X-OVER・NIGHT

（クロスオーバーナイト）

［第39話］

おかしなこと

TAKAHARA HIDEKI 高原 秀己

エレクトロニクス産業を冷めた目で眺めてみると，不況のあおりで収益力がガタガタになっていることはいうまでもないが，マーケティング戦略を見ても，おかしな点が少なからず目につく。

●ハイビジョンとワイド画面テレビ

たとえば消費者にとって不要な商品を売りつけようとする姿勢が強すぎる。代表的なケースがハイビジョン受像機だろう。

一昨年からの家電不況の理由のひとつとして挙げられているのが，このハイビジョン受像機の立ち上がりの悪さだという。

そんなもの，売れなくて当然である。

決定的に見落とされているのが，100万円という価格の高さだ。メーカーはどうしても，コストパフォーマンスの観点から高いか割安かを判定する悪癖がある。しかし，バブル破綻後の消費動向は，予算重視のうえでの内容追求。いくらいいものでも，高すぎるものは売れない。

おそらく，大画面テレビがすっかり定着したものだから，さらに上位のテレビが発売されれば買う人は多い，という思い込みがメーカーにはあるのだろう。

だが，100万円は明らかに高すぎる。秋には80万円のが出るそうだが，やはりまだまだ高い。

それでも内容で圧倒していればいいとしよう。だがハイビジョン版のソフトなど，ろくに放送されていない。性能はともかく，本来の機能が発揮できない状態なのだ。

どうして当たり前のことがわからないのだろうか，と改めて不思議がっていると，秋からの目玉商品は，ワイド画面テレビなのだそうだ，という記事を目にした。

やれやれ。

●MD

その点，MD（ミニディスク）は，久々にわくわくする期待の家電製品だ。事実，すでに買った人によると，すごく便利な商品だという。規格争いの綱引きもあるのだろうが，こういう期待の高い商品は早々に市場を拡大して価格の引き下げをするようメーカーに努力してもらいたいものだ。

しかし，このMD，実に不思議な売り方をしている。録音ができない機種しか売っていない企業があるのだ。というよりも本家のソニーにしたところで，録音再生型機種に劣らないウエイトで再生専用機に期待しているようで，「ソフトの充実を進めています。もう300タイトルを超えましたよ」なんていっている。

何か勘違いしてないだろうか？

CDがこれだけ普及しているなかでMDが新たに登場したのは，録音可能なランダムアクセス型メディアであるからにほかならない。ソフトを買って聴きたい人はCDのほうがいいに決まっている。しかも音質はCDのほうがいいそうだ。

とすれば，ソフトを用意しているヒマなどあるのなら，とりあえず録音再生機をガンガンと売りまくって，ある程度普及してから再生機のマーケティングをするのが筋なのではなかろうか？

というわけで，早く５万円程度の据え置き型録音再生MDプレーヤーを商品化してくれー，と絶叫したいところ。

でもいまは10万円弱もするのだから，しばらくは無理なのかなあ？

●意味不明な言葉

やたらと難しい言葉が先行しているのも，エレクトロニクスメーカーの奇妙なところ。「ダウンサイジング」くらいコンセプトがはっきりした言葉なら問題はないのだが，「コンピュータストラテジー」になってくると，かなり「？」。「ソリューション」や「コラボレーション」だのとなると，もう末期的な気分すらしてくる。

おそらくは広告代理店あたりが中間に入って，メーカーの趣味的担当者とともに意味不明言葉を仕立てあげるのだろうが，そんなのはプレゼン段階で封印して結界の紙封でも貼っておいていただきたい。便利でいい機械なりシステムを利用者に提供することが重要なのであって，言葉遊びはどうでもいいのである。

注：確たる理解や評価もなく嬉しそうにそういった言葉を受け売りするマスコミも責任大であることはいうまでもない。

●マルチメディア

逆に一見，簡単そうな言葉なのだが，実際にはちゃんとした使われ方がされていなくて，非常に安易に流行語となっているのが「マルチメディア」だろう。

朝日新聞社から発売の『朝日キーワード』の最新版では，「映像と音声をコンピューターに取り込むこと」ときわめてあっさりとした，しかし突き放した説明がしてある。

各種の宣伝や解説を読んでみても，これまでは容量が大きすぎて親和性が薄かった音声，映像とコンピュータ，データ通信とをミックスする趣旨のものがほとんどだ。だが，音声や映像とコンピュータや通信との掛け算が「マルチメディア」というのは，言葉の乱用に思えて仕方がない。

本来の意味を考えてみると「マルチ」な「メディア」が扱える技術，ということなのだから，そう珍しがる必要はない。そもそも処理速度や容量が飛躍的に高まれば，映像や音声が扱えるのは当然のことだ。だから，ただ単にCD-ROMなり高速通信機能があるからといって「マルチメディア型パソコン」だなどとPRしてはおこがましい。

やはり最終的に，幅広い種類の情報を適確に，しかも簡単に扱えるソフト面での整備が必要だろう。それなしにハードだけ辻褄合わせをして，ソフトはアメリカの会社から持ってくればいいや，なんていう姿勢では，進歩はない。

ちなみに，10年ほど昔のことだが，パソコンテレビX1とかX1turboという製品があった。当初からテレビ画像をパソコン画面に取り込んだり，のちにはデジタルデータとしてハンドリングする機能が備えてあった。いまにして思えば，大変な先見性だ。

ただし。

そのメーカーとユーザーは10年分のリードを，資産として十分に活用しているのだろうか？　その点からも目はそむけないでほしい。

illustration：Haruhisa Yamada

第136部　シューティングゲームコアシステム作成法(4)

●投稿テーマ＆プログラム大募集

THE SENTINELへの投稿が，完全にワシントン条約で保護されてもいいくらいに減り，一家離散＋借金背負って路頭に迷いそうな今日この頃。ここで一発，「THE SENTINEL WORLD」と題して読者からの意見を大々的に募集しようと考えています。基本方針としては，以下について読者からの意見，プログラムを募集します。

1)　○○がほしい＆××はどうや

S-OS用のツールでこんなのがほしい，こんなツールや言語があれば面白いんじゃないか，といった要望を募集します。最低限の約束ごととしては，必ずどの機種のS-OSでも動く，ということを前提としてください。ある程度の例外は認めますが，各機種のハードウェアを叩くようなものは控えましょう。既存のものに因らわれず，オリジナリティあふれる投稿を待ってます。

2)　ご意見するぞ！

「THE SENTINEL WORLD」で紹介されたテーマやツールの問題点を暴いたり，自分のやりたいことをいいたい放題しゃべってもらうコーナーです。

3)　テクニック自慢

S-OSのワンポイントテクニック，Z80のプログラミングテクニックなど，自分の知っているちょっと役立つ情報を読者の皆さんに公開してしまおう，というものです。

4)　テーマプログラミング

毎月なにかしらのテーマを提示し，読者の皆さんでそのテーマに取り組むものです。採用の基準は，短い，多機能，速いの三拍子，やっぱり質実剛健なものを選びます。

いままで，S-OSに参加できずにやきもきしていた人，特に「プログラムはちょっと苦手だなあ。でもアイデアだけは人に負けないぞ」という人は，思いついたアイデアを文書にまとめ，どしどしお寄せください。「こんなことをやってみては？」といった提案も大歓迎です。

もちろん「俺はこんなものを作っているぞ（または作ろうとしている）」なんてのもOKです。皆さんの活動状況も積極的に紹介していきたいと思います。基本的には，この「THE SENTINEL WORLD」のコーナーで発表する予定ですが，ある程度本数がまとまればページ数を確保してどど～んと紹介する可能性もあります。

もちろん引き続き皆さんからの投稿をお待ちしていますので，遠慮なくTHE SENTINELに働きかけてください。

●これからのTHE SENTINEL(3)

それでは，栄えある第1回目のテーマを発表します。テーマは「シューティングゲームコアシステムを完成させよう」です。これは，タイトルそのまんま。今月で尻切れトンボになってしまう「シューティングゲームコアシステム作成法」の新しいサービスコールを，読者の皆さんに手伝ってもらうものです。サービスコールは，連載で書かれていたものでも，独自に考えたものでもかまいません。

「ふ～ん，なんか面白そう」「これぐらいならやれそう」と思ったら即実行です。「ゲームシステムを作るのはいいけど，なんで他人の尻まで拭ってやらなきゃいかんのだ」とお思いの方，そういうあなたは，ぜひ，オリジナルシステムを作ってみませんか。こちらもプログラムだけではなく，より具体的な仕様書でもかまいません。

また，比較的新しい読者のために「シューティングゲームコアシステム作成法」の記事（1993年3～5月号）のコピーサービスを先着30名様にかぎり行います。

応募方法は，官製ハガキの裏に住所，氏名，年齢，電話番号，使用機種名，「コピーサービス希望」と明記のうえ，下記の宛先まで，お送りください。

Oh!X編集部　「シューティングゲーム
　　　　　　コアシステム作成法」コピー係

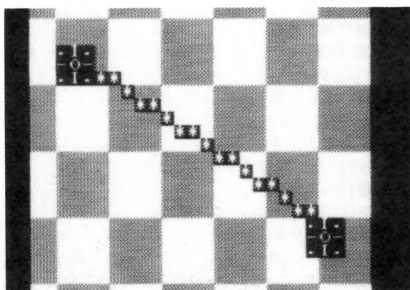
なお，コピーサービスを行うのは，1993年3～5月号のみで，ほかの号についてのコピーサービスは行いません。ご了承ください。

1993■インデックス

全機種共通
S-OS"SWORD"要

シューティングゲーム
コアシステム作成法(4)

Sakamaki Katsumi
坂巻 克巳

残念ながらシューティングゲームコアシステム作成法も今回で最終回。坂巻氏は改めてシステム作成の難しさに気づいたようです。時間的な制約とはいえ，非常に残念な結果に終わってしまいました。

第1部完?

長らく連載を中断していましたが，今月からめでたく再開……ということもなく，小見出しにあるとおり一応の区切りをつけさせてもらいます。ここ数カ月間，時間がまったく取れず（あくまで個人的なことですが)，これから先も状況はあまり好ましくありません。めどが立たないのに思わせぶりなセリフを吐くのは，あまり気持ちのいいものではありませんから。

ということで，読者の皆様には誠に申し訳ありませんが，事実上今回が最終回となります。僕としても途中で投げるのは，非常に悔しい思いがあります。状況が許せば，また新たに挑戦していきたいのですが……どうなることやら。

では，弁解はこれぐらいにして，最終回らしくいままでのまとめと，偉そうにゲームシステムを作成するためのウンチクを語っていきます。

表示ルーチンの先へ

いままで3回にわたってこの連載で完成させたものは，キャラクタの表示とその移動ルーチンです。

そして，ここから先の予定として，キャラクターの管理を予定していました。ひと口にキャラクターを管理する，といってもピンとこない人のためにも，具体的にどのようなことをするのか説明していきます。

1)　登場キャラクターの制御

まずは画面内を動き回るキャラクターの制御です。キャラクターが自機の場合であれば，

・キー入力によってキャラクターの移動を行う

・必要であればキャラクターを書き換えてアニメーション処理を施す

・特定のキーが入力されたら弾を発射する

などがプログラムで記述され，これが，敵キャラクターであれば，

・あらかじめ決められたデータに従ってキャラクターを移動させる。もしくは決められたプログラムでキャラクターを移動させる

・もし，弾を発射させるならその動作を行う

・必要であればキャラクターを書き換えてアニメーション処理を施す

ということをプログラムで記述します。普通のゲームでは，こういったキャラクターごとのメインルーチンをもっています。

で，肝心のどのキャラクターが画面に登場して，どのルーチンを使うかを判定するにはどうしたらいいのでしょうか。

これは，まず，画面に登場できるキャラクターのワークエリア([適当なサイズ×登場可能なキャラクター個数］バイト）を用意します。そして，1キャラクターごとに適当なサイズに割り振ったワークの中身を，

・キャラクターの種類（0だったらワークは未使用，それ以外だったらキャラクター番号とする)

・表示座標

・移動データアドレス

・弾の発射カウンタ

⋮

というように，プログラマが意味づけを行うのです。全体のイメージとしては，

敵1のキャラクター番号
敵1の各種情報
敵2のキャラクター番号
敵2の各種情報
敵1のキャラクター番号
敵1の各種情報
敵Nのキャラクター番号
敵Nの各種情報
⋮
(敵キャラクターの出現最大個数まで)

という具合にずらずら〜っとワークが並ぶことになります。

そうして，敵のキャラクターのメインルーチンでは，まず，敵キャラクターの種類を調べ，それぞれのメインルーチンへ制御を移します。メインルーチンでは，ワークに格納された各種情報をもとに，移動を行ったりさまざまな行動を起こすのです。

2)　キャラクターどうしの当たり判定

今度は，当たり判定です。最初に単純に座標による当たり判定を考えます。

まず，キャラクターどうしの表示位置の差を求めます。そして，それぞれのキャラクターの大きさを考慮して当たり判定有効範囲を決めます。最後に表示位置の差が，判定有効範囲内に収まっていれば，そのキ

ャラクターどうしが接触した，と判定されるわけです。この判定を自機側のキャラクターの数×敵側のキャラクターの数だけ繰り返せば，画面に登場しているすべてのキャラクターどうしの当たり判定を行うことができます。

普通の当たり判定はこんな感じですが，ここで，連載の最初のほうを思い出してください。今回作成しようとしたシステムでは，仮想画面が3ページ分用意されています。そこで，ページごとにどの種類のキャラクターを表示させるか決めてしまうことで，当たり判定は多少簡単になります。

まず，VRAM1に敵キャラクター，VRAM2に自機を表示するとします。シューティングゲームでの当たり判定は，たいていこの自機と敵機との間で行われるものです。すると当たり判定は，自機が表示されている範囲のVRAM1側になにかキャラクタが置かれているかどうか，で判定できます。

また，逆の方向，つまり敵機側から自機との当たり判定をすると考えれば，敵側のキャラクターが表示されている範囲に該当するVRAM2の内容を調べるだけですむのです。しかし，この方法ではある問題があります。それは，自機の弾側から敵機との当たり判定を考えた場合，当たり判定に引っ掛かったときに敵側のどのキャラクターと衝突したか知るすべがないのです。

これは，敵側から調べても同じことです。この場合はしようがないので，キャラクターが衝突したと判定されたあと，さらに先ほど説明した座標による判定を行う必要があります。

こう説明すると「なんだ，仮想画面を使っても結局座標による当たり判定が必要ならかえって遅くなるんじゃないか」と考えるかもしれません。

しかし，ゲームが進行している間の当たり判定の状況は，キャラクターどうしの衝突が起きない時間のほうが圧倒的に多い，と考えられるのです。つまり，トータルでみれば後者の仮想画面を調べ，必要に応じて座標による当たり判定を行うほうが速い，ということになります。これは，当たり判定に必要なループ回数をカウントしていけば，理解してもらえるでしょう。

敵側から判定を行うか，自機側から判定を行うかは，ゲームによって違うでしょう。最終的にシステムとしては，両方から対応できるようなルーチンを用意すればいいことになります。

だいたい，以上のようなことをサポートするだけでシステムが完成しそうです（なんかだいぶ連載1回目にいっていたことと違う気がするけど)。となると，結構簡単にできそうな気がするなあ。う～ん，誰かやりません？

七転び八起き

さて，記事の冒頭で述べたように，今回は，見事失敗に終わりました。これが，敗者のいい見本です。暗い夜道には気をつけます。笑って石でも投げてください。

ただ，ひとついいたいのは，自由度の高いシステムとは汎用性をもち細分化されたルーチンの集合体だ，ということです。極端な話，当たり判定の仕様などを確定せず，今回の連載で作った表示ルーチンだけのほうが自由度が高い，といえるのです。当たり判定やキャラクター管理などは，作るゲームによってコロコロ変わるものなので，いちいち制約されていたのでは，システム以上のものが作れません。

いちばんいいのは，やはりゲームごとに最適化された表示ルーチン，当たり判定システムを作ることです。でも，やっぱりゲームごとに専用ルーチンを作るのは，無駄な努力のように思えますし，たいてい似通ったものになります。だったら，まとめてしまえ，ってな具合に誰もが考えるでしょう（僕もそうだった)。

ただ，こういったシステムは個人で使う分にはなんの問題もありません。細かい仕様が自分の頭の中にあるのですから当然ですね。ところが，いざほかの人が使うとなると手に負えないことのほうが多いのです。これは，いままでOh!MZ,Xで発表されてきたものを見ればわかります。MAGICしかり，SYSYTEM-7Bしかり，もちろんX68000の世界でも例外ではありません。

現実には，これらのかなりしっかりしたシステムでも，それを使いこなしたアプリケーションプログラムは皆無に近い現状です。ま，これは，他人のシステムを使える人は，同等のシステムを構築できる，または構築できるくらいのパワーをもった人，と解釈できないこともないのですが。逆に初心者などには，システムを使うための敷居がかなり高いところにある，といえなくもありません。このへんのバランスは非常に難しいですね。

汎用性を追求すれば，結局，自分ですべてを管理して使いこなす必要があるし，専用システムにしてしまっては作られるものがかぎられてきてしまうでしょう。

これからシステムを作ろうとしている人，現在制作中の人も，いま一度自分のシステムを見直してみませんか。ぜひ，よりよいものを目指してください。

1986

■86年1月号
第13部　リロケータブルのお話
第14部　FM音源サウンドエディタ
■86年2月号
第15部　S-OS "SWORD"
第16部　Prolog-85入門(1)
■86年3月号
第17部　magiFORTH発表
連載　Prolog-85入門(2)
■86年4月号
第18部　思考ゲームJEWEL
第19部　LIFE GAME
連載　基礎からのmagiFORTH
連載　Prolog-85入門(3)
■86年5月号
第20部　スクリーンエディタE-MATE
連載　実戦演習magiFORTH
■86年6月号
第21部　Z80TRACER
第22部　magiFORTH TRACER
第23部　ディスクダンプ＆エディタ
第24部　"SWORD" 2000 QD
連載　対話で学ぶmagiFORTH
特別付録　PC-8801版S-OS "SWORD"
■86年7月号
第25部　FM音源ミュージックシステム
付録　FM音源ボードの製作
連載　計算力アップのmagiFORTH
特別付録　SMC-777版S-OS "SWORD"
■86年8月号
第26部　対局五目並べ
第27部　MZ-2500版S-OS "SWORD"
■86年9月号
第28部　FuzzyBASIC発表
連載　明日に向かってmagiFORTH
■86年10月号
第29部　ちょっと便利な拡張プログラム
第30部　ディスクモニタDREAM
第31部　FuzzyBASIC料理法<1>
■86年11月号
第32部　パズルゲームHOTTAN
第33部　MAZE in MAZE
連載　FuzzyBASIC料理法<2>
■86年12月号
第34部　CASL & COMET
連載　FuzzyBASIC料理法<3>

1987

■87年1月号
第35部　マシン語入力ツールMACINTO-C
連載　FuzzyBASIC料理法<4>
■87年2月号
第36部　アドベンチャーゲームMARMALADE
第37部　テキアベ作成ツールCONTEX
■87年3月号
第38部　魔法使いはアニメがお好き
第39部　アニメーションツールMAGE
付録　"SWORD" 再掲載とMAGICの標準化
■87年4月号
第40部　INVADER GAME
第41部　TANGERINE
■87年5月号
第42部　S-OS "SWORD" 変身セット
第43部　MZ-700用 "SWORD" をQD対応に
■87年6月号
インタラプト　コンパイラ物語
第44部　FuzzyBASICコンパイラ
第45部　エディタアセンブラZEDA-3
■87年7月号
第46部　STORY MASTER
■87年8月号
第47部　パズルゲーム碁石拾い
第48部　漢字出力パッケージJACKWRITE
特別付録　FM-7/77版S-OS "SWORD"
■87年9月号
第49部　リロケータブル逆アセンブラInside-R
特別付録　PC-8001/8801版S-OS "SWORD"
■87年10月号
第50部　tiny CORE WARS
第51部　FuzzyBASICコンパイラの拡張
第52部　X1turbo版S-OS "SWORD"
■87年11月号
序論　神話のなかのマイクロコンピュータ
付録　S-OSの仲間たち
第53部　もうひとつのFuzzyBASIC入門
第54部　ファイルアロケータ＆ローダ
インタラプト　S-OSこちら集中治療室
第55部　BACK GAMMON
■87年12月号
第56部　タートルグラフィックパッケージTURTLE
第57部　X1turbo版 "SWORD" アフターケア
　　　　ラインプリントルーチン
特別付録　PASOPIA7版S-OS "SWORD"

1988

■88年1月号
第58部　FuzzyBASICコンパイラ・奥村版
付録　石上版コンパイラ拡張部の修正
■88年2月号
第59部　シューティングゲームELFES
■88年3月号
第60部　構造型コンパイラ言語SLANG
■88年4月号
第61部　デバッギングツールTRADE
第62部　シミュレーションウォーゲームWALRUS
■88年5月号
第63部　シューティングゲームELFES II
第64部　地底最大の作戦
■88年6月号
第65部　構造化言語SLANG入門(1)
第66部　Lisp-85用NAMPAシミュレーション
■88年7月号
第67部　マルチウィンドウドライバMW-1
連載　構造化言語SLANG入門(2)
■88年8月号
第68部　マルチウィンドウエディタWINER
■88年9月号
第69部　超小型エディタTED-750
第70部　アフターケアWINERの拡張
■88年10月号
第71部　SLANG用ファイル入出力ライブラリ
第72部　シューティングゲームMANKAI
■88年11月号
第73部　シューティングゲームELFES IV
■88年12月号
第74部　ソースジェネレータSOURCERY

1989

■89年1月号
第75部　パズルゲームLAST ONE
第76部　ブロックゲームFLICK
■89年2月号
第77部　高速エディタアセンブラREDA
特別付録　X1版S-OS "SWORD"<再掲載>
■89年3月号
第78部　Z80用浮動小数点演算パッケージSOR
　　　　OBAN
■89年4月号
第79部　SLANG用実数演算ライブラリ
■89年5月号
第80部　ソースジェネレータRING
■89年6月号
第81部　超小型コンパイラTTC
■89年7月号
第82部　TTC用パズルゲームTICBAN
■89年8月号
第83部　CP/M用ファイルコンバータ
■89年9月号
第84部　生物進化シミュレーションBUGS
■89年10月号
第85部　小型インタプリタ言語TTI
■89年11月号
第86部　TTI用パズルゲームPUSH BON!
■89年12月号
第87部　SLANG用リダイレクションライブラリDIO.LIB

1990

■90年1月号
第88部　SLANG用ゲームWORM KUN
特別付録　再掲載SLANGコンパイラ
■90年2月号
第89部　超小型コンパイラTTC++
■90年3月号
第90部　超多機能アセンブラOHM-Z80
■90年4月号
第91部　ファジィコンピュータシミュレーションI-MY
■90年5月号
第92部　インタプリタ言語STACK
■90年6月号
第93部　リロケータブルフォーマットの取り決め
第94部　STACK用ゲームSQUASH!
第95部　X68000対応S-OS "SWORD"
特別付録　PC-286対応S-OS "SWORD"
■90年7月号
第96部　リロケータブルアセンブラWZD
■90年8月号
第97部　リンカWLK
■90年9月号
第98部　BILLIARDS
■90年10月号
第99部　ライブラリアンWLB
■90年11月号
第100部　タブコード対応エディタEDC-T
■90年12月号
第101部　STACKコンパイラ

1991

■91年1月号
第102部　ブロックアクションゲームCOLUMNS
■91年2月号
第103部　ダイスゲームKISMET
■91年3月号
第104部　アクションゲームMUD BALLIN'
■91年4月号
第105部　SLANG用カードゲームDOBON
■91年5月号
第106部　実数型コンパイラ言語REAL
■91年6月号
第107部　Small-C処理系の移植
■91年7月号
第108部　REALソースリスト編
■91年8月号
第109部　Small-Cライブラリの移植
■91年9月号
第110部　SLANG用NEWファイル出力ライブラリ
■91年10月号
第111部　Small-C活用講座（初級編）
■91年11月号
第112部　Small-C活用講座（応用編）
第113部　MORTAL
■91年12月号
第114部　Small-C SLANGコンパチ関数

1992

■92年1月号
第115部　LINER
■92年2月号
第116部　シミュレーションゲームPOLANYI
■92年3月号
第117部　カードゲームKLONDIKE
■92年4月号
第118部　オプティマイザO80実践Small-C講座(1)
■92年5月号
第119部　COMMAND.OBJ実践Small-C講座(2)
■92年6月号
第120部　COMMAND.OBJ2実践Small-C講座(3)
■92年7月号
第121部　関数リファレンス実践Small-C講座(4)
■92年8月号
第122部　ワイルドカード実践Small-C講座(5)
第123部　グラフィックライブラリ GRAPH.LIB
■92年9月号
第124部　O-EDIT&MODCNV
■92年10月号
第125部　SLENDER HUL実践Small-C講座(6)
■92年11月号
第126部　EDIT実践Small-C講座(7)
■92年12月号
第127部　MAKE実践Small-C講座(8)

●編集部へのメッセージ

今月号の特別企画について

いちばん良かった記事

興味のなかった記事

これから載せてほしい記事内容

本誌以外にお読みのパソコン雑誌

期待している新作ソフト：
推薦理由：

最近買って気に入ったソフト：
推薦理由：

付録ディスク「秋祭りPRO-68K」で，いちばん興味があったのはどれですか？

あなたの愛機は(所有機種に○印をつけてください)　ない
X1(マニアタイプ,C,D,F,G,twin) X1 turbo(model 10,20,30,40,II,III,Z,ZII,ZIII)
MZ-(80K/C, 1200, 700, 1500, 80B, 2000, 2200, 2500, 2861)
X68000(初代,ACE,PRO,PROII,EXPERT,EXPERTII,SUPER,XVI,Compact,HD)
X68030(CZ-500/510,300/310) その他　MIDI楽器(　)
FD(　基)　TAPE　QD　HD(　MB)　MO　プリンタ(　)

年齢　歳　パソコン歴　年　男・女　プレゼントNo.

キリトリ線

振替用紙

←点線から、きれいに切り取ってご使用ねがいます。

通常払込料金
加入者負担

払込票

口座番号	東京 1 - 29307
加入者名	ソフトバンク株式会社
金額	億 千 百 十 万 千 百 十 円
払込人住所氏名	※
備考	受付局日付印

切り取らないで郵便局にお出しください。

記載事項を訂正した場合は、その箇所に訂正印を押してください。

切り取り線

通常払込料金
加入者負担

払込通知票

口座番号	東京	1 - 29307	金額	億 千 百 十 万 千 百 十 円
加入者名	ソフトバンク株式会社		料金	払込み / 特殊 円
払込人住所氏名	※ （郵便番号　　）		備考	
			受付局日付印	

各票の※印欄は、払込人において記載してください。

この払込通知票は、機械で使用しますので、下部の欄を汚さないよう特に御注意ください。また、本票を折り曲げたりしないでください。　（郵政省）

切り取り線

【定期購読のご案内】

●定期購読のお申し込みは、この郵便振替用紙のみとさせて頂き、銀行振込・現金書留によるご入金は、ご遠慮下さい。

●受付締切は、

1日発売　：　発売日前月10日振込
8日発売　：　〃　15日振込
15・18日発売：　〃　25日振込
です。

〈例〉 4月1日発売（Oh！PC 4月15日号）の場合、お振込の締切は3月10日です。
締切に間に合わなかった場合は、自動的に次号からの発送となります。
なお、すでに発売されているもの、また、お振込が締切に間に合わなかった月号のものは、定期購読ではお求めになれません。書店でご購入ください。

●定期購読誌のお届けは書店発売日より遅くなりますのでご了承下さい。

「発売日一覧」

◇毎月1・15日発売
Oh！PC

◇毎月18日発売
Oh！X
Oh！FM TOWNS
C MAGAZINE
Oh！Dyna
月刊PC

◇毎月8日発売
月刊情報処理試験
LANTIMES
THE WINDOWS
DOS／Vmagazine
UNIX USER

●月刊情報処理試験は93年1月号より定期購読料金を改訂させていただいております。お申し込みの際はご注意下さい。

切り取り線

切り取らないで郵便局にお出しください。

この欄は、加入者あての通信にお使いください。

送り先

お名前　フリガナ　　お電話

ご住所　フリガナ　〒

定期購読申込書

□Oh！PC	年間	（23回）	12,880円	（新規／継続 NO.　）
□Oh！PC	6ヶ月	（12回）	6,720円	（新規／継続 NO.　）
□Oh！X	年間	（12回）	7,200円	（新規／継続 NO.　）
□Oh！FM TOWNS	年間	（12回）	7,440円	（新規／継続 NO.　）
□C MAGAZINE	年間	（12回）	11,760円	（新規／継続 NO.　）
□Oh！Dyna	年間	（12回）	9,120円	（新規／継続 NO.　）
□月刊 PC	年間	（12回）	7,800円	（新規　）
□月刊情報処理試験	年間	（12回）	9,360円	（新規／継続 NO.　）
□月刊情報処理試験	6ヶ月	（6回）	4,680円	（新規／継続 NO.　）
□LANTIMES	年間	（12回）	17,760円	（新規／継続 NO.　）
□THE WINDOWS	年間	（12回）	11,760円	（新規／継続 NO.　）
□DOS／Vmagazine	年間	（12回）	9,360円	（新規／継続 NO.　）
□UNIX USER	年間	（12回）	11,760円	（新規／継続 NO.　）

〔備考欄〕

この払込通知票は、機械で使用しますので、下部の欄を汚さないよう特に御注意ください。また、本票を折り曲げたりしないでください。　（郵政省）

愛読者プレゼント

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1993年10月18日の到着分までとします。当選者の発表は1993年12月号で行います。また，雑誌公正競争規約の定めにより，当選された方はこの号のほかの懸賞には当選できない場合がありますので，ご了承ください。

コナミ
☎03(3432)5526

悪魔城ドラキュラ

X68000用 5″2HD版

9,800円(税別)　3名

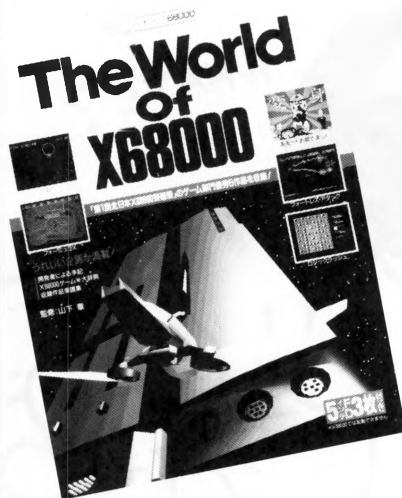
1年ぶりのコナミの新作。次々と繰り出される美しいグラフィックや細部まで行き届いた演出など，期待にたがわぬ大作です。

電波新聞社
☎03(3445)6111

The World of X68000

X68000用
5″2HD版

4,800円(税込)

5名

24ページでも紹介しているゲームディスクつきの本。芸術祭優秀5作品を収録しています。

EAビクター
☎03(5410)3111

湯飲み

非売品　5名

2カ月続きで紹介した「コットン」。あのティータイムに使われている湯飲みです。WILLOWでも食べながらどうぞ。

アルファレコード
☎03(3455)1791

ステッカー

非売品

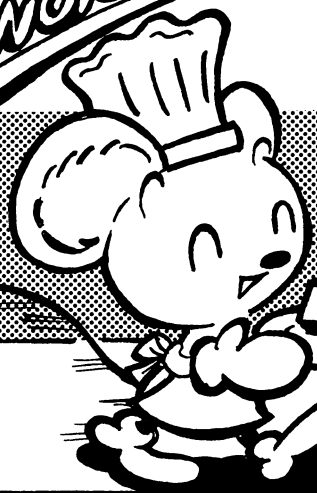
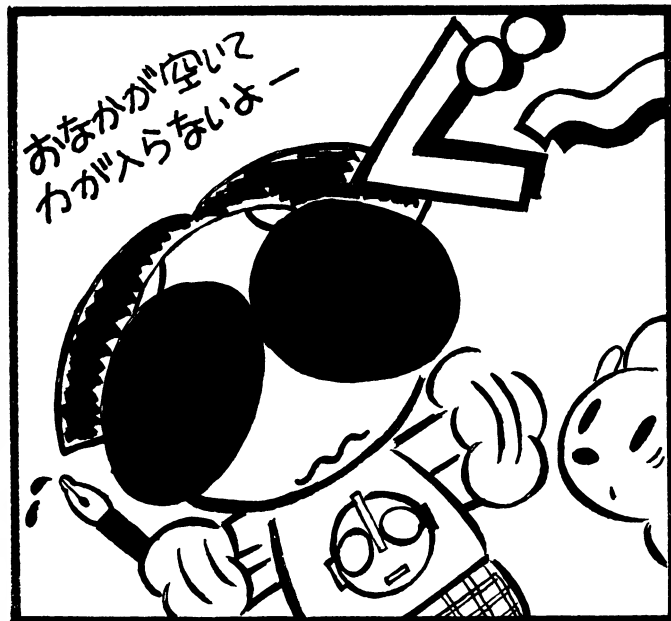
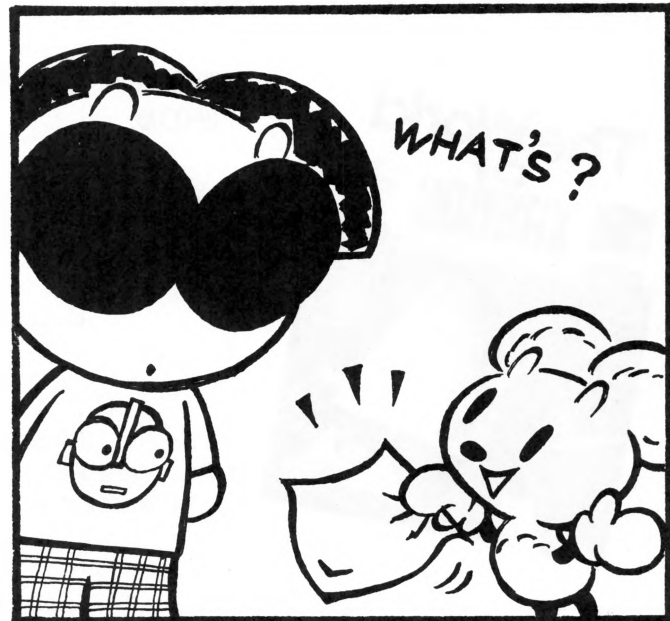
30名

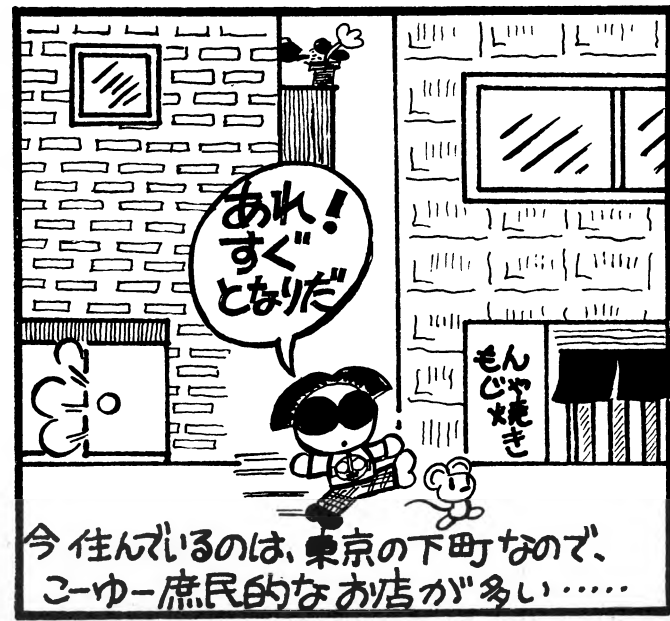
いまや「世界的有名人」となったスーパーマリオのCDのステッカー。

8月号プレゼント当選者

1ヴェルスナーグ戦乱 (大阪府)春藤大輔 (広島県)馬場徹太郎 (福岡県)秋友謙二 2幻影都市 (東京都)佐藤 遊 (神奈川県)安沢光男 (長野県)宮島雅史 3X68000 Free Software Book (北海道)小林正輝 田坂克明 (栃木県)吉見英人 (静岡県)坂井国彦 (愛知県)市古信二 4X680x0 libc (東京都)越智由浩 (京都府)片岡 学 (兵庫県)林田誠司 (広島県)秋山欣之 (長崎県)山下 寛

(敬称略)

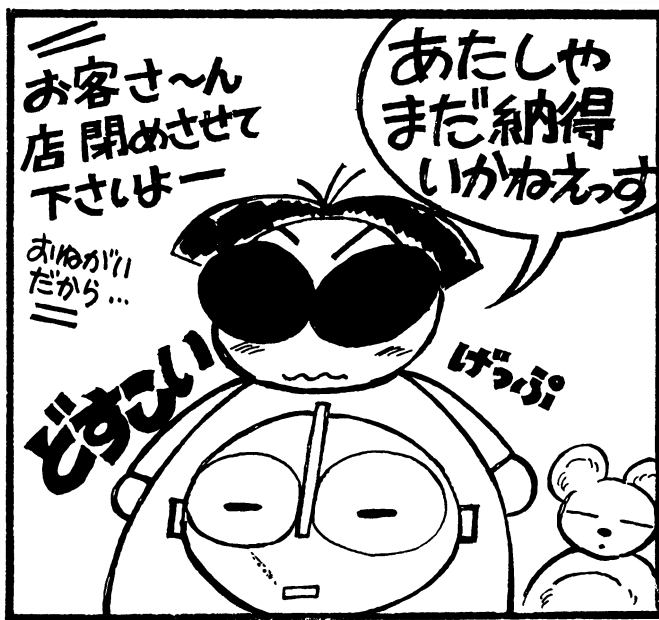
以上の方々が当選しました。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。

ANOTHER CG WORLD
CHEESE CAKE
CHEESE OMLET
CHEESE TOAST
KYOKO'93

おなかが空いて
力が入らないよー

WHAT'S?

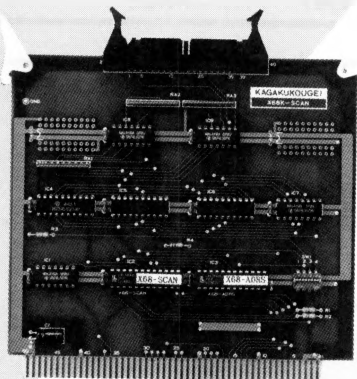
もんじゃ焼きの店OPEN
すぐそこ
LETSGO
ちなみにもんじゃ焼きとは、
お好み焼きをもっと
ドロドロに水っぽくした
ものでござんす。

あれ!
すぐ
となりだ
もんじゃ焼き
今住んでいるのは、東京の下町なので、
こーゆー庶民的なお店が多い……

今回のCGデータ

手前のキャラクタの物体数47
うちメタボール数6
背景作成にZ'sSTAFF PRO-68K
キャラクタは別々に計算して合成
使用ソフトはC-TRACE

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

電脳絵師のためのスキャナボード
X68K-SCAN
科学工芸研究所

X68K-SCAN

科学工芸研究所はスキャナ用パラレル入力ボード「X68K-SCAN」を発売する。

本製品は，X68000/X68030でエプソン「GTシリーズ」スキャナを使うためのパラレルボードである。「GTシリーズ」は，RS-232Cでの取り込みができたが，転送速度に問題があった。それが本製品を使うことで，かなり軽減できる。具体的には，512×512ドット，65,000色の取り込みを実行した場合，RS-232C，19,200bpsの場合は7分17秒かかったものが，61秒で読み取りが完了した（同社の調べによる）。

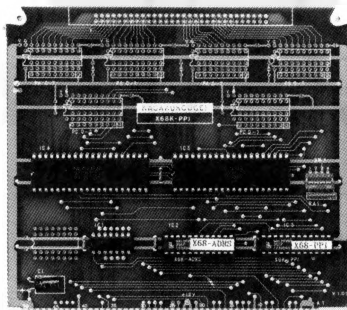
同梱されているものは，ボード，接続ケーブル，サンワード「MATIER」上でパラレル入力を可能にするソフト(5/3.5インチ同梱，インタフェイスルーチンはライセンスフリー，ソフトそのものはコピーフリー)，取り扱い説明書(全回路図，ソースリスト公開）である。

対応スキャナは「GT-1000」「GT-4000」「GT-6000」「GT-6500」「GT-8000」である(「GT-6500」はシリアル・パラレルボードが必要)。なお，シャープ製パラレルボード「CZ-6BN1」との互換性はない。

価格は29,000円（税別)。

〈問い合わせ先〉

㈱科学工芸研究所　☎03(5385)4651

自作派御用達8255コンパチボード
X68K-PPI
科学工芸研究所

X68K-PPI

科学工芸研究所は8255コンパチのインタフェイスボード「X68K-PPI」を発売する。

本製品は，グラフィックや音楽と同期してモーターなどを動かすなど，汎用的な8255の出力を提供するインタフェイスボードである。

同梱されているものは，48ビットI/Oボードと取り扱い説明書。説明書の中ではボードの全回路図とGALの論理を公開している。ボードにはμPD71055（8255コンパチ）が2個搭載され，入出力用バッファICが搭載できるエリアが8ビット×6個分用意されている。

また，入出力コネクタやバッファIC，プルアップ抵抗などは添付されないため，各種部品の用意とそれにともなうハンダづけ作業が必要になる。

X68030にも対応。

価格は22,000円（税別)。

〈問い合わせ先〉

㈱科学工芸研究所　☎03(5385)4651

新携帯情報ツール
PI-7000(英語バージョン)
シャープ

PI-7000(英語バージョン)

シャープは新携帯情報ツール“Expert Pad”「PI-7000」を米国で発売した。

本機は，シャープと米国アップルコンピュータ社の共同で，Newtonテクノロジーをベースとして開発された商品である。

特徴としては，ペンオペレーションで操作し，筆記体/ブロック体文字および図形などが認識できる。PIM(Personal Information Management)機能では，名刺管理やスケジュール管理が行える。ほかにも世界時計や電卓などのアプリケーションも内蔵されている。

また，オプションのFAXモデムによりファクシミリへ送信可能，電子メールとして送受信が可能になり，オプションのリンクソフトでMacintosh，IBM PC互換機との通信が可能となる。ICカードスロットルを1基搭載し，今後ICアプリケーションカードを順次供給していく予定。

CPUは32ビットRISC CPUを搭載し，640KバイトのRAMを内蔵している。

本体のサイズは，111.5mm（幅)×181.5mm（奥行)×27.8mm（厚さ）とほぼ新書サイズと同じくらいの大きさである。重さも単4電池を4本入れた状態で約440gと

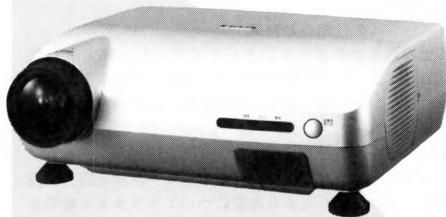
携帯性に優れている。

価格は899USドル。

〈問い合わせ先〉

シャープ㈱☎043(299)8210,06(621)1221

液晶ビジョン
XV-T2Z
シャープ

XV-T2Z

シャープは高輝度液晶ビジョン「XV-T2Z」を発売した。

本機は,112,320画素の液晶パネルを3枚採用,水平解像度400本の高画質を実現した。200Wメタルハライドランプの採用により,スクリーンの4隅までより均一な明るさで,鮮明な画像を楽しめるようになった。また,映画ソフトなどの横長画面の場合は,映像の上下を黒帯でマスキングして,ワイド映像を鮮明に映し出す。

設置方法は,天吊り,棚置き,据置きなど,生活スタイルに合わせて自由に選ぶことができる。そして,色あわせ調整の必要がないため簡単に設置できる。

さらに,アンプ・スピーカーを内蔵しているため,ビデオデッキやビデオカメラと接続するだけで大画面を楽しめるようになっている。短焦点電動ズーム(フォーカス)レンズの採用により,10畳の広さで100型が楽しめる。ビデオ入力端子は2系統,S映像入力端子は1系統,モニター出力端子を1系統装備。

なお,ライトアップリモコンの採用により,暗い部屋でも簡単に操作ができるようになった。

価格は600,000円(税別)。

〈問い合わせ先〉

シャープ(株)☎043(299)8210,06(621)1221

ファクシミリ電話機
UX-T1
シャープ

UX-T1

シャープは,ファクシミリもコピーもできるファクシミリ電話機「UX-T1」を発売した。

本機は,B4サイズでの送受信が可能で,「中間調32階調(写真モード)」により写真やイラストの濃淡を表現する機能も充実している。FAXの操作も音声・光・液晶ディスプレイなどを使って操作をガイドしてくれる。また,最近増えているFAX情報サービスから情報を取り出すときにも,タイトルを登録し,そのあとに行った操作を1件記憶させることが可能。次回からはワンタッチで前回と同じ操作を自動的に行い,情報が引き出せる。タイマ設定により,夜間にはノンコールでの受信が可能である。

コピー機能も本体の読み取り部分を取りはずし,B4ハンドコピーが行えるため,ノート綴じの原稿もコピーが可能になった。

本体の大きさは,297mm(幅)×210mm(奥行)×95mm(高さ)。

価格は65,000円(税別)。

〈問い合わせ先〉

シャープ㈱☎043(299)8210,06(621)1221

スキャンコンバータ
XVGA-1V
マイコンソフト

XVGA-IV

マイコンソフトはスキャンコンバータ「XVGA-1V」を発売した。

本機は,パソコンの画像出力を家庭用テレビで見たり,ビデオに録画したりできるユニットである。具体的にはパソコンの24〜35kHzまでのアナログRGB信号をNTSC標準のビデオまたはSビデオ信号(15kHz)に変換する。

同梱されているものは,本体,ACアダプタ,ビデオ接続ケーブル,S端子ケーブル,IBM-PC用RGB変換ケーブル(入力用1本,出力用1本),15ピン・アナログRGBケーブル,取扱説明書など。

接続可能な機種は,X68000シリーズのほか,PC-9801,FM TOWNS,Macintoshシリーズ,DOS/Vパソコンなどがある。

価格は69,800円(税別)。

〈問い合わせ先〉

電波新聞社出版販売部　☎03(3445)8201

INFORMATION

Silicon Graphics EXPO'93
日本シリコングラフィックス

9月号で紹介したビジュアルパーソナルコンピュータ「INDY」を発表したシリコングラフィックスが,3次元コンピュータグラフィックの祭典「SiLicon Graphics EXPO'93」を,横浜アリーナにて,'93年10月13,14日の2日間にわたって行う。会場では新製品デモンストレーションや国内外のゲストによるシンポジウム,CGフィルムショー,体感マシンを利用したバーチャル・リアリティの体験,マルチメディアデモンストレーションなどが行われる。

開場時間は10:00〜17:00で,入場は無料。新横浜駅,綱島駅から無料送迎バスも出ている。

〈問い合わせ先〉

「SILICON GRAPHICS EXPO'93」運営事務局　☎03(3221)6463

FILES

Oh!X

このインデックスは，タイトル，注記――著者名，誌名，月号，ページで構成されています。文化祭や運動会の季節ですね。ほかにもなにかと忙しいけど，食べ物がとっても美味しいから許しちゃう。いっぱい食べて，ファイト！

参考文献

I/O　工学社
ASCII　アスキー
コンプティーク　角川書店
C Magazine　ソフトバンク
テクノポリス　徳間書店
電撃王　主婦の友社
POPCOM　小学館
マイコンBASIC Magazine　電波新聞社
My Computer Magazine　電波新聞社
LOGIN　アスキー

一般

▶目指せ!!　ゲームクリエイター
ゲームクリエイターを目指す読者のために，ゲーム作家養成講座学校を取材し，メーカー側の話を聞く。最後にログイン流クリエイターの4つの掟を掲げるぞ。――編集部，LOGIN，15号，203-219pp.

▶電網幼稚園
通信は始めたけど，まだフリーソフトをダウンロードしたことがない人に，ダウンロードの実際をやさしく伝授。――編集部，LOGIN，15号，254-257pp.

▶THE NEWS FILE!
「買ったその日に使える一体型マック登場！」アップルの「LC 520」発売のニュースや，レーザーアクティブ発売日決定のニュースなど，パソコン関連の話題。――編集部，LOGIN，16・17号，30-37pp.

▶電網幼稚園
初心者ネットワーカーのための連載ページ。今回は夏休み特別企画「パソコン通信の歴史を振り返ってみよう」だ。同時に移動体通信でアウトドアでの通信を試みる。――編集部，LOGIN，16・17号，286-289pp.

▶光磁気ディスクMO
データがだんだん大きくなる昨今，フロッピーに代わるリムーバブルメディアとしてMOが大きく成長している。その用途と魅力を探る。――編集部，マイコンBASIC Magazine，9月号，39-49pp.

▶新製品 Flash News
シャープの構造化BASIC関数ポケコン「PC-E650」など，パソコン周辺機器の新発売情報。――編集部，マイコンBASIC Magazine，9月号，78-82pp.

▶Bug太郎のプログラム・タイム　その9
今回のテーマはずばり格闘ゲーム。ゲームセンターをにぎわすあのゲームに挑戦！　というわけで，まずは多関節のキャラクターをどうやって表示するかを解説する。――谷裕紀彦，マイコンBASIC Magazine，9月号，88-89pp.

▶BASICプログラミング講座
今月は読者から投稿されたプログラムを紹介し，アルゴリズムを解説する。投稿は素数の検索，微分係数の基礎など数学の教育プログラムだ。――森誠一朗，マイコンBASIC Magazine，9月号，94-98pp.

▶BASIC MAGAZINE NEWS
話題の3DOの発売日決定のニュース，ハイビジョン対応「HI-TENボンバーマン」の大会開催やストIIターボTVCMの裏話など，ゲーム関連の話題が満載。――編集部，マイコンBASIC Magazine，9月号，179-183pp.

▶パソコン雑誌大研究
ひと口にパソコン雑誌といってもいろんな種類がある今日このごろ。POPCOM編集部がライバル各誌を分類し，自らの雑誌ができるまでを紹介する。各ゲーム誌編集長による麻雀大会レポートつき。――編集部，POPCOM，9月号，43-53pp.

▶新鮮良品館
熱い夏を思いっきり楽しむための新製品のバラエティグッズを紹介する。指先から6mも先まで水をとばせる水鉄砲，新ジャンルの電子楽器など。――編集部，POPCOM，9月号，118-119pp.

▶HDD&MOはこれだ！
大容量記憶メディアを必要とする大型ソフトが増えてきた。MOやリムーバブルHDの製品評価を中心に，用語解説やMO普及の事情，記憶メディアの今後について触れる。――編集部，ASCII，9月号，177-192pp.

▶近未来情報端末II
いよいよ我々の前に姿を現し始めたペンコンピュータ。カシオとタンディによるZoomer，アップルとシャープの共同開発のNewtonの機能を研究し，「ポストノートパソコン」の行方をさぐる。――編集部，ASCII，9月号，275-288pp.

▶バカパパのモノを買い物
今月は無差別級物品図鑑の巻。ノンジャンルだからこそ現れたというおかしなグッズがたくさん登場。卓上扇風機やバーチャルTVなど。――バカパパ，ASCII，9月号，332-333pp.

▶最新のハードディスク事情
市販のソフトウェアもハードディスクの搭載を前提にしたものが多い昨今，どんなハードディスクを買えばいいのか。各社主要ドライブを比較。――支倉槙人ほか，My Computer Magazine，9月号，35-71pp.

▶今月の注目
X68000対応のスキャンコンバータ「XVGA-Iv」や，ヒューレット・パッカードのインクジェットプリンタ「DeskJet300J/505J」などを紹介。――編集部，My Computer Magazine，9月号，86-93pp.

▶ビジネスマンのための情報管理術
シャープのハイパー電子手帳の活用講座。今月は「ハイパー関数プログラムカード」の機能と応用例を紹介する。――塚田洋一，My Computer Magazine，9月号，194-197pp.

▶ゲーム・プログラミング入門
アイデア＋デザイン＋プログラミングテクニックから成り立っているゲーム作り。その過程で必要となるアイデアの骨格作りやプログラムへの反映のさせ方などを，6人の筆者が解説する。――土方嘉徳ほか，I/O，9月号，9-39pp.

▶ムッシュの仕組みと制御
「世界最小のロボット」としてギネスにも認定された「ムッシュ」とは，光に向かって加速しながら進むマイクロマシンの一種である。その制御システムを開発者が解説する。――宮沢修，I/O，9月号，54-60pp.

▶スーパーコンピューティング入門
カオスとフラクタルを研究するシリーズ第8回。今回は我々の身の回りにある地図についてフラクタル操作を応用してみる。――林智雄，I/O，9月号，144-145pp.

X1/turbo/Z

X1シリーズ

▶くらげくんなすびくん
2人で遊ぶX1用対戦ゲーム。くらげとなすびがゲートを目指して競い合うという謎のシチュエーションがグーだ。――中村理，マイコンBASIC Magazine，9月号，136-137pp.

X1turboシリーズ

▶SLASH
星がきれいな落下型パズルゲーム。斜め方向に同じ色の星を4つそろえて点数を稼ごう。――ぷるぅ　ふぁいやぁ　そふと，マイコンBASIC Magazine，9月号，138-140pp.

X68000

▶最新ゲーム徹底解剖
「信長の野望・覇王伝」の東海，甲信越地方の攻略法解説や「大航海時代II」から世界の港湾紹介など。――編集部，LOGIN，15号，136-139，144-147pp.

▶X68030新聞
セガの傑作シューティング「コットン」がX68000に登場するニュースと，シャープのドローイングツール「EasydrawSX-68k」を紹介する。――編集部，LOGIN，15号，234-235pp.

▶NEWSOFT Radar!
スピタル産業の連射機能つきジョイパッドにX68000用LIMITED EDITIONが登場。その他“多摩21くらしの祭典”にアートディンクからオリジナルゲーム「TAMA21」が出展されたニュースなどを取り上げる。――編集部，LOGIN，16・17号，12-15pp.

▶NEW SOFT
自分で作ったロボットが自分で組んだプログラムで動く！　エレクトリックシープの「ロボットコンストラクションR.C.」をプログラムの話を中心に紹介。――編集部，LOGIN，16・17号，26pp.

▶X68030新聞
新発売のカラーイメージスキャナー「JX-325X」を紹介。新たにスタートした「クリゲー紹介!!」コーナーでは，X68000の名作ソフトウェアを毎号1作ずつ取り上げる。――編集部，LOGIN，16・17号，264-265pp.

▶未確認クリエイターズ

読者からの投稿に与えられるログイン大賞。その発表に向けてノミネート作品を紹介する。X68000用「世界征服セット」など3作品。——編集部，LOGIN，16·17号，290-295pp.

▶CPUパワーアップ工事

WINDOWSの普及とともにCPUのパワーアップが人気を集めている。そこでパワーアップパーツの情報を提供するのがこのページ。今月はX68000Compact XVI改造マシン「RED ZONE」を取り上げる。——編集部，マイコンBASIC Magazine，9月号，68-71pp.

▶SUPER SOFT HOT INFORMATION

93年8月に発売される各機種用のゲームソフト115本のニュースをまとめた別冊。X68000用では「ロボットコンストラクションR.C.」「クレイジークライマー/クレイジークライマー2」ほか2作を紹介。——編集部，マイコンBASIC Magazine，9月号，別冊11-12pp.

▶BOM BOM

邪魔なブロックを爆弾の爆発で焼却し，宝石を横一列に並べて獲得する落下型パズルゲームだ。爆弾の連鎖反応は快感だぞ。——知ったか庄ちゃん，マイコンBASIC Magazine，9月号，141-143pp.

▶Small Volleyball

2人のキャラクターがボールを打ち合うバレーボールゲーム。コンピュータ相手でも人間同士の対戦でも遊べるぞ。——高橋秀之，マイコンBASIC Magazine，9月号，144-146pp.

▶ナックルヘッズ

ナムコ初の格闘ゲームより，シルバの曲をX68000＋GS音源用にアレンジしてお届けする。——牧田竜也，マイコンBASIC Magazine，9月号，156-158pp.

▶速報ネメシス'90改

一時期発売無期延期とも報じられた「ネメシス'90」がいよいよX68000で発売決定。その詳細をレポートする。——編集部，マイコンBASIC Magazine，9月号，202-204pp.

▶SUPER SOFT INDEX

話題の新作情報コーナー。X68000用には「餓狼伝説」と「宝魔ハンターライム2」が紹介されている。今後の新作発売スケジュールも掲載。——編集部，コンプティーク，9月号，29-49pp.

▶HOW TO WIN

光栄のシミュレーションゲーム「項劉記」「信長の野望・覇王伝」「大航海時代II」，コナミの新作「悪魔城ドラキュラ」などの攻略法を伝授する。——編集部，コンプティーク，9月号，51-97pp.

▶GAME PARADISE

X68000用の発売も予定されている「項劉記」など新作ゲーム攻略のアドバイスとレビュー。——編集部，電撃王，9月号，31-51pp.

▶Denngekiパソコン

X68000用熱血格闘アクション「餓狼伝説」や電波新聞社の「クレイジークライマー/クレイジークライマー2」などの新作ソフト情報。——編集部，電撃王，9月号，68，75pp.

▶ニューゲームREPO!!

「ダーク・オデッセイ」や「項劉記」「CALIII」「コットン」などX68000用各種新作ソフトウェアを一挙に紹介。——編集部，テクノポリス，9月号，10-36pp.

▶ゲームの達人

X68000版の発売が近い「項劉記」のレビューをはじめとした，各機種用最新ゲームのレビュー記事。——編集部，POPCOM，9月号，58-61pp.

▶AV STRASSE

X68030で快適なマルチタスク環境を実現する，マイクロウェア・システムズの「OS-9/X68030」を取り上げる。日本語FEP VJE-γを搭載。——編集部，ASCII，9月号，297-300pp.

▶TBN GAME

10年ぶりにお茶の間に現れたナムコの名作「リブルラブル」を2ページにわたってレビュー。——上野利幸，ASCII，9月号，326-327pp.

▶FREE SOFTWARE INDEX

ここ1〜2カ月の間に主要ネットにアップロードされたソフトウェアを，編集部が厳選して掲載するページ。X68000用テキスト行分割ツール「TCP.X」ほか。——編集部，ASCII，9月号，369-375pp.

▶なんでもQ&A

「シャーペン.X」でイメージデータをきれいに印刷する方法はあるか？　など，X68000を活用するうえで生じる疑問に答える。——シャープAVCシステム事業推進室，9月号，225-227pp.

▶GCCで学ぶX68ゲームプログラミング

ゲーム作成の連載。ゲームの仕上げにかかる。ビジュアルの演出効果をもたらす「半透明機能」について，作成過程を描写しながら解説していく。——吉野智興，C Magazine，9月号，118-124pp.

ポケコン

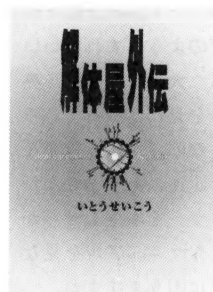
PC-E500

▶Squash

壁を相手にボールをひたすら打ち返すアクションゲーム。しだいに壁がせまってくるのがハードだ。——今坂光太，マイコンBASIC Magazine，9月号，147p.

▶JUMP MAN（改良バージョン）

島々をジャンプしてゴールまでたどりつくという，ワンキーアクションゲームの改良版。——近藤紀之，マイコンBASIC Magazine，9月号，148-149pp.

新刊書案内

解体屋外伝
いとうせいこう著
講談社刊
☎03(5395)3622
四六判　352ページ
1,600円（税込）

解体屋にはデ・プログラマーとルビが振られている。反対語は洗濯屋でウォッシャーだ。洗濯屋は洗脳する人で，解体屋は洗脳を解除する人。洗脳というとすぐ「セミナー」や「統一教会」をネタにしたなと思うところであるが，いとうせいこうは前作「ワールズ・エンド・ガーデン」でもこれらの職業を登場させている。本書もただ流行にのって解体屋を主人公にしたわけではなく，前々から温めていた題材が世間の流行と重なってしまったと見るのが正しいだろう。

この解体屋と洗濯屋集団の物語はハイテクを使わないサイバーパンクとして見ることも可能だ。人間の心へのアプローチにテクノロジーではなく言葉のテクニックを使うというだけで，物語の展開やノリはサイバーパンク的エンターテイメントに近い。用語の作り方や精神世界の表現方法など多分に電脳的でサイバーパンク風になっている。ただ，エンターテイメントとして流してしまうにはあまりにテーマがディープであり，時代を鋭く突き過ぎている。日本にはアメリカで生まれたハイテクを使ったサイバーパンクよりもこちらのほうが似合いそうだ。特に，企業が社員教育に洗濯屋を使い始めるくだりなどは，いまの時代をちょっとデフォルメしただけで，実際，セミナー的手法はかなり根を下ろしている。さらに，薄く洗脳して消費者を誘導しようという言葉は街中で散見できるほど，いまの時代は人々の不安を軽くあおる暗示と薄い洗脳に満ちている。だから，本書で際立つのも，洗濯屋と解体屋の物語より，解体屋が語る洗脳と解体の仕組みであり，そのクールさであり，解体屋の戦いぶりだ。人の精神をどこまで解体できるか，って言葉の限界で溺れ死ぬようなテーマにまで深く沈む必要もあるまい。

「暗示の外に出ろ。俺たちには未来がある」というキーワードが頻出する。暗示の中にいるぬるま湯的心地好さへの警鐘だ。鋭いお話である。（K）

電脳激動
坂村健著
日刊工業新聞社刊
☎03(3222)7131
四六判　226ページ
1,900円（税込）

ひとつのモノ（コンピュータ）がドラスティックに安くなったことにより社会構造の変化をもたらし，世の中に強いインパクトを与える構図，それが電脳激動。そのなかで起きたと考えられるいくつかの出来事，たとえばダウンサイジング，オープンシステム，根本の技術の独占化などについて考察している。そして，これからもしばらく続くであろう電脳激動について，今後注意して対応すべき知的所有権問題などいくつかの点を，国際的な視点から述べている。坂村氏の提唱するTRONを含めた独自技術の必要性を時代の流れという観点でわかりやすく説明した本である。

株式会社ダーウィン商事
赤池学著
みずき出版刊
☎03(5684)6201
四六判　209ページ
1,500円（税込）

本書は，基本的にビジネスバイブルである。「株式会社ダーウィン商事」，この会社は創業35億年で，社員たちは現在300万。これまで地球上に生きてきた生物の種の数である。その長い歴史のなかで，成功してきた生物たちは，個性や特徴を最大限に活かして競争し，「生態的優位」を獲得した。それに敗れた生物は消えていった。そんな社員（生物）たちの20の物語に，現実の企業での事象との共通性を見つけ出し，そこからある種のビジネス格言をおもしろく導き出している。ビジネス書としてだけでなく，生物の雑学書としても楽しく読める本である。

QUESTION and ANSWER

バージョンの違うZ-MUSICで作られた曲を新しいシステムで演奏させると曲の前にBEEP音が鳴るのですが，なぜですか。

千葉県　石井　英一郎

曲の前にBEEP音が鳴るというのは，BEEP音が鳴ってちゃんと音楽が演奏されるということでしょうか。これはZPDファイルがみつからない場合くらいしかないはずですが。

それ以外の場合にはデータの中のどこかでエラーが発生しているので，コンパイルしてエラー箇所を確認してください。

ZMUSIC -C ファイル名

とすることでコンパイルが行われます。このときには詳細なエラー表示がありますので，そのメッセージを参照して修正してください。ちなみにZP.RやCOPYコマンドではエラー箇所がわかりません。

さて，Z-MUSICシステムではZMSレベルでの互換性が保証されています。ただし，これは正しく記述されているデータの演奏に関してのみ当てはまり，万一，誤ったデータを演奏しようとしたときには同じ症状を示すとは限りません。

確かに古いバージョンで演奏できていたデータが新しいバージョンでそのまま演奏できないこともあります。理由は簡単で，その元々のデータに誤りがあったからです。最近のZ-MUSICはver.1.1などよりもエラーチェックが厳しくなっています。以前は指定しても動作に影響のない無意味なコードは無視されていたのですが，これがチェックされるようになったのです。

これは全体的なバグチェックを容易にし，データ内の無駄を省くための変更でもあります。これまでは指定したつもりなのに動作していないコマンドなどがあってもチェックできていませんでした。これでは無意識のうちにデータにバグを入れてしまう可能性があります。

ちなみにこういった症状の場合，“TIE COMMAND ERROR”であることがほとんどです。この場合，指定された行のどこかに無意味な“&”があるはずですので削除しておいてください。

とりあえず，誤解のないデータを作るという意味でも，無駄をなくすという意味でも，Z-MUSIC ver.1.1で作成されたデータをお持ちの方は，付録ディスクに収録された新しいバージョンで正常に演奏されるかどうかを確認して，必要ならばデータを修正したほうがよいでしょう。

なお，ZMDデータになっている場合は動作が保証されていませんので，必ずZMSデータを用いるようにしてください。

SX-WINDOWのアイコン登録，あれよくわかりませんねぇ。そのような方はいませんか？　ひとつ登録するとほかのファイルが全部それになっちゃうとか。どのようにすればいいのでしょうか。

大阪府　越智　亮

マニュアルを見れば載っているはずのことなのですが，ほとんど具体的な記述はありませんから解説しておきましょう。そもそもせっかくのGUIなのにいちいちマニュアルを見なくては使えないようでは困りますよね。

さて，肝心のマニュアルを見てみても，アイコンパターンの変更に関しては詳しく解説されているにもかかわらず，ユーザーが新しくアイコンを追加するような場合についてはほとんど解説されていません。

アイコンの管理なのだから，アイコンメンテ.Xだろうとあたりをつけても，越智さんのようにほかのアイコンまで変わってしまったり，すでに定義されていたアイコンを破壊してしまったりすることが多いのではないでしょうか。

SX-WINDOWでは，アイコンは独自の管理番号を持っています。アイコンメンテではその番号で定義されたアイコンに対してファイル名や実行ファイル，アイコンパターンを指定します。昔のビジュアルシェルを使っていた人はファイル名と直接対応するものではないということに十分注意してください。

さらに，アイコンメンテ.Xは現在すでに登録されているアイコンの保守を行うものですので，新しくアイコンを定義することはできません。ここでいう新しいアイコンとは「新しいファイル名」に対応するアイコンを意味します。マニュアルに解説されているのはすべて「新しいアイコンパターン」ですので注意してください。

まるでお役所仕事のような仕様ですが，ユーザー定義のアイコンの作成はいったいどこで行われるのでしょうか？　これは実際にはアイコンリスト.Xで行われます。ちなみにアイコンリストはアイコンメンテの「アイコンリスト」ボタンで呼び出すこともできます。

ここでポップアップメニューから「新規」を選択することにより，ようやく新しい種類のアイコンをエディットすることができるようになります。

機能が複雑になりがちなアプリケーションならともかく，基本ツールでは名前から直感的に機能がわかるようなファイル名でないと困ります。特にこれらは相互に呼び出されるので，わかりにくいからといってユーザーが勝手に名前を変更するわけにもいきません。ツールごとの機能の分担なども適切とは思われません。むしろかなり悪い見本といえます。SX-WINDOWプログラマの皆さんは真似をしないようにしてください。

MOのディスクを入れ替えて他社製のドライブで読み書きしようとするとき，ドライブの相性によっては読み取りが不可能なものがあると聞きました。どうなっているのでしょうか？　また，ハードディスクモード（コパルなど）でデータを書いたMOのメディアを他社の光磁気ディスクモードで読み込むことは可能でしょうか？　また，正しく書き込むことは可能でしょうか？

静岡県　岡田　徹

図1　アイコンメンテ.X

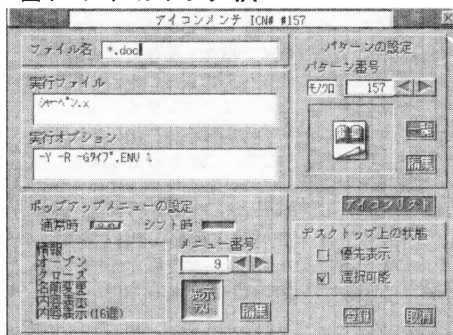
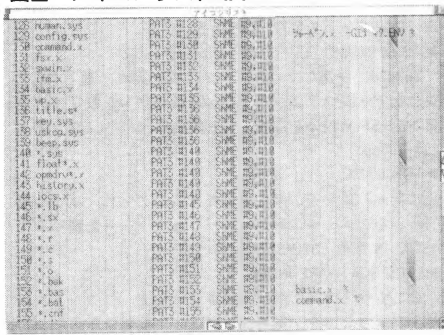



図2　アイコンリスト.X

MOというのは一応規格に沿った製品ですので，ドライブの物理的な問題で互換性が損なわれているものというのはほとんどないと思われます。

ただし，最近は旧IBM製品準拠の1800回転ドライブとの互換をとっていないドライブもありますので，旧IBM製品とそういったものについては完全には保証されません。また，ナカミチなど独自規格で倍容量MOを製作しているものについても今後の互換性は保証されないと考えていいでしょう。無論倍容量のものは通常のドライブでは読めません。現状では128Mバイト，3000回転，さらにいえばソニードライブ準拠でIBMフォーマットの場合の互換性がもっとも優れています。

ただし，そのような問題のあるドライブはあまり一般的ではありませんので，実質的にはほとんど問題ないと考えていいようです。ちなみに，9月号で試用したドライブではすべて問題なくMOメディアの相互使用が可能で，モードによらずデータ互換がとれています。

むしろ，ドライブの問題というよりも，MOメディアを気をつけて選ぶようにしたほうがよいでしょう。編集部で試用したなかでは帝人のMOメディアは3600回転での書き込みに耐えられないのではないかという疑いがもたれています。データをよそに持っていくという場合には，やはり3600回転対応と明記されたメディア（三菱化成だけか？），またはそれなりに実績の確認されたものをおすすめします。

次にハードディスクモードでの使用ですが，これらはSCSI機器がどう認識されるかという問題だけで，メディア自体に実質的な違いはありません。

通常の読み書きではまったく問題がないと考えて結構です。

ただし，メディアのフォーマット時に微妙に容量が変わってしまいますので，ハードディスクモードのドライブでフォーマットされたメディアと光磁気ディスクモードのドライブでフォーマットされたメディア間では，MO 2台を直結した場合にDISKCOPYコマンドでディスクの複製を行うことはできません。まあ，これで困る人が多いとは思いませんが……。

容量の微妙な違いはフォーマットプログラムの違いでも発生します。しかし，フォーマットさえされていればデータの読み書きではまったく関係ありません。

MOって1パーティションで使えば使用中にメディアを交換できるのですか。以前5インチMOではできないとありましたが。あとフォーマットはFORMAT.XかHDフォーマット.Xで行えるのでしょうか（マニュアルを読めばいけそうなんだけど，念のため)。それから，特定のメーカーのメディアはX68000ではフォーマットできないと聞きましたがどうなのでしょうか。あと，中野氏の記事の中に「ソニードライブではイジェクトが……」とありますが，これってオートイジェクトのことなのでしょうか。つまりソフトさえ作ればMOでもFDと同じくオートイジェクトできるってことですか。

大阪府　森　秀樹

3.5インチMOではFDと同じように起動中にMOメディアの交換ができます。5インチMOの場合はわざわざディスク交換をチェックしてはじいていたような形跡が認められます。

1パーティションでなくとも，パーティション構成さえ同じであればMOメディアの交換ができます。

ただし，ハードディスクモードで接続されている場合など，システムがリムーバブルメディア（交換可能メディア）として認識していない場合には交換の際にブレイクキーを押すなどしてディスクバッファをクリアする必要があります。ディスクキャッシュなどを使っている場合もバッファをクリアするようにしたほうがよいでしょう。

次にフォーマットですが，FORMAT.XでもHDフォーマット.Xでもどちらでもかまいません。SCSI対応のフォーマットコマンドであれば，システム付属のものだけで特に専用プログラムなどは必要ありません。

5インチMOの場合はMOメディアの出荷時に物理フォーマットされているのが常識だったわけですが，3.5インチMOの場合にはメーカーによっては物理フォーマットされていないメディアが販売されていることがありました。

よって，以前はMacintosh用のMOを買ってきて接続しても付属のメディアがフォーマットされていなかったために使用できないという事態もあったようです。

以前のFORMAT.XではSCSI機器の装置初期化が異様に速かったことを覚えている人はいないでしょうか。そもそも大容量のSCSI機器は製品のチェックを兼ねて出荷前に基本的な物理フォーマットを行っていることが多かったのです。昔のFORMAT.Xでは，ひととおりの物理フォーマットがなされていることを前提としていたため，まったく物理フォーマットされていない機器に対しては無力でした。

最近のFORMAT.Xは物理フォーマットにも対応しているようですので特に問題はないでしょう。そもそも，いまどきそんな変なMOを入手することのほうが難しいという話もありますが。

ただし，MOではありませんが，SCSIハードディスクのなかには1ブロックがHuman68kでサポートされているサイズ（512バイト）以外のものがありますので，ごく稀にフォーマットできないものもあるようです。MOの場合は512バイトと規定されていますので大丈夫です。

最後の質問ですが，イジェクトというのはもちろんオートイジェクトのことです。MOはSX-WINDOWなどではイジェクトボタンでもオートイジェクト可能です。プログラムでイジェクトを行う場合は，記事にあったとおりSCSIコマンドのSTART STOPに引数2を指定して実行してください。

（中野　修一）

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を挙げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに解答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので電話番号も明記してくださいね。

宛先：〒103　東京都中央区日本橋浜町3-42-3
ソフトバンク株式会社出版部
Oh!X編集部「Oh!X質問箱」係

FROM READERS TO THE EDITOR

読書の秋，食欲の秋，スポーツの秋，芸術の秋，さまざまな秋がある中で，皆さんは，なにをして過ごしてるかな。落ちゆく枯れ葉が舞う並木通りを歩いていると妙に感傷的になってくる。今日はちょっと近くの美術館にでも行ってみよう。

◆今までiocslibのLINE()がややこしくて使いづらかったのですが，丹さんのbasライクな使用方法を見て，こういう使い方があったのか！　と目からウロコが落ちたようです。

蟻馬　章生(20)宮崎県

◆C言語を学んでいる。簡単なプログラムは作れるようになったけど，ゲームでも作ってみようと思うと，とたんになにをしていいのかわからなくなる。どうすればいいか教えてくれ。

船木　亮(19)神奈川県

特集としてのCは久しぶりです。どうもお待たせしました。読んでいろんなことに気がついた人や，さらにはまった人などいろいろといるようですが，はりきってプログラムを組んで投稿してください。

◆田村さんの“しつこくアクセサリ”を読んで，初めてFEPを「フェップ」と読むことを知りました。「エフイーピー」といままで読んでました。しかし，「フェップ」にはかなり抵抗を感じますね。

大野　隆士(22)沖縄県

FEPは最初から「フェップ」と読んだので気にならなかったんだけど，9月号の“3DO”，皆さんはなんて読みますか。たぶん「スリーディーオー」だと思うけど，「スリードゥ」とか，あと「サンド」なんて読み方はいかがでしょう。だめですか，でも「サンド」って結構気にいったんですけど。

◆ついにX68030を買いました。初代機が出たときからコツコツ貯めてやっと手に入れました。友人のPROII（HDなし）に寄生する生活ともこれでおさらばです。あとは触る時間がもう少しあればなぁ。さあ，明日はなにをしようかな。幸せ，幸せ。

河野　敏弘(22)広島県

初志貫徹，ここに至るまでずいぶん時が過ぎましたが，これからはついに自分だけのX68030ですね。その幸せを私にも分けて。

◆金銭的問題から1年間ハードディスクなしでやってたけど，やっと買えた。よかった。今度はMIDI楽器も欲しい。でもDATも欲しい。

金田　和也(16)北海道

どんどん充実していく環境，ただ満足いく環境をそろえるのはむずかしいですね。

◆親父にRED ZONEが欲しいので多少金銭的に助けてほしいと相談したら，RED ZONEのあまりの仕様のあやしさ，そして低価格に，親父は宗教がからんでいるあやしい通販と誤解？　僕にX68030を買ってくれました。パソコンショップ満開とはいったい……。次は，あやしい6ボタンパッドとかを売りだしてほしいですね。それにしても40,000円のプリンタを買ってくれない父が，なぜX68030を買ってくれたのでしょうか？　満開おそるべし（ネーミングに問題があるのかな）。

太田　貴道(20)静岡県

お父さんの機嫌がよくてよかったですね。ちがうって。ま，なににしろいまあなたの手にはX68030があるんだから。いいなあ。

◆今年の春に，私はなんと夢の中でデバッグをしてしまった。ゲームのプログラムを作っているとき，なかなか取れないバグが出て，残りは明日やろうと眠ったら，夢の中でX68000の前に座り，「これはファイル読み込みのサイズが違っている」とつぶやく自分がいるではないですか……おどろいて，目を覚ましたのでまさかと思ってX68000を起ち上げてみたら本当にそこが間違っていたのです。

千葉　浩貴(20)宮城県

私の周りにもデバッグで苦労している人がちらほら。夢でのデバッグ作業はあっても，現実での解決をみた話は聞かないなあ。ラッキーじゃないですか，毎回こうだといいんですけどね。

◆「リブルラブル」を買いに行ったとき，ソフトを手に取って見たら，専用パッドが入ってないのでよく見たら，「バブルボブル」だった。

阿部　祐三(23)神奈川県

ありがち，ありがち。急いで買って家に帰ってあらびっくりみたいな。買う前に気がついてよかった，よかった。

◆先日，大学の教育用計算機室のワークステーションのディスプレイから突然「ボン」という音と共に煙が出てきました。近くにいた人たちと一緒に，まるでマンガの出来事のようだ，と騒いでしまいました。

竹原　充(20)栃木県

煙が出ただけだったからよかったものの，火を吹いていたら，マンガのような出来事といってられなかったでしょうね。でも，そんなにマシンを酷使してたの？

◆私がX68000を買った当時は，メモリが2Mバイトあれば動かないソフトはなかった。時は流れて，ふと気がつくと「メモリーが2Mバイトしかありません」のメッセージが……うちの娘だって5年もたてば大きくなるのだから，毎日電気食べてるメモリだって成長したっていいのになあ。こういうところだけは律儀なんだから。

野原　賢次(32)埼玉県

メモリよりはCPUのほうが電気をいっぱい食べてるような気がするので68030に成長してくれないかなあ……無理ですよね。

◆かなり前からだが，うちのディスプレイはときどき，「パチッ」という音と共に画面が1フレームだけ消える。その横にいる我が愛機「ステラ」。う～ん，けなげだ。

坂田　宗隆(18)大阪府

今，この原稿を書いているディスプレイも「パチッ」という音と共に画面が消えてしまう。あっ，この瞬間にもまた……うりゃ，「バン」(ディスプレイの側面を叩く音)，さてお仕事しよっと。

◆生活に疲れた……というより，山積みされて

▲岡村　直也　兵庫県
すでに家族にも彼氏を紹介したとのこと。今度は，お姉さんの彼氏も登場するかな。でも結婚して，お姉さんが登場しなくなると寂しいな。

▲小川　伸輔　宮城県
コンピュータの場合，年をとるにつれて老いていくのではなく，幼児退行していくのか。するとSUPERも，気持ち幼く見えてきたのは私だけ。

いる課題にちょっとイラついている。夏休みといえば……はあ，考えたくもない。毎日，マウスを持ってゴリゴリと設計だなんて！　しかし，もう少しで完成。終わったら，次の課題。少し寝不足。外は雨。人並みの楽しみはないのであろうか？　　　　中村　学(21)福岡県

外が雨ならそんなときにこそ課題はすませて，晴れた日にパッと外で気晴らししようよ。

◆7月18日(日)，大相撲名古屋場所，せっかく曙・若・貴の巴戦になったのに兄弟対決は実現せず。う〜ん残念。夜は選挙速報のTVを見ましたが，野球と2元中継にする意義はあるのだろうか？　　　　春名　義行(26)兵庫県

選挙速報のTVだけど，意義はともかくあの画面を見た瞬間「なんじゃこりゃ！　見て，見て」と騒いだ私，Windowsの画面を見ているようで，なにかまぬけで好きだったんですけど。

◆今年生まれて初めて「帰省ラッシュ」というものを経験しました。もう二度と経験したくないですね。　　　　藤原　彰人(23)岡山県

でも，1年に1回くらいは家に帰って，親孝行しようよ。あと，わりと朝早い時間だったり，始発を待ったりすると大丈夫かもしんない。車だとチョットわかんないけど。

◆僕の部屋にもあります。ウゴウゴデッキ（笑）。ウゴウゴ・ルーガのおかげでビデオデッキをもう1台買う決心がつきましたね。8月17日からの生放送も成功してほしいです。

桜井　高史(18)千葉県

あいかわらずシュール君は好きなんだけど，最近ウゴウゴ・ルーガにごぶさたしている。生活のリズムが……。

◆大学も2年目にしてようやくゆとりが持てるようになった。自己無力感をごまかすために演じる一生懸命さではなく，好きだから夢中になれるひたむきさ。そいつを背負って自然体で生きていきたい。今までせかせか生き急いできたもんな。　　　　加藤　伸一(21)神奈川県

人がどのように生きようとも時間だけは過ぎていく，そのなかでどうやって生きていくかは自分が決めること，頑張ってください。

◆欲は出さないつもりでいても，ついつい見てしまうプレゼントのコーナー。宝クジとどっちが先に当たるだろう？　藤原　睦夫(24)岡山県

宝クジは毎週出ているから買い続ければそっちのほうがいいかもよ。でもOh!Xも毎月買ってね。

◆友人と表紙の色を賭けて2カ月連続で負けてます。なんとかしてください。

鈴木　勇(19)北海道

そうですか。では，そんなあなたのためにそっと教えてあげましょう。11月号の表紙はですね……おっと，ここで書いたら友達にもわかっちゃうから，ナ・イ・ショ。

◆僕の悩みを聞いてください。僕はペチャパイなのですが，どうすれば豊かな胸になれるのでしょうか。　　　　大島　大介(17)北海道

◀相沢　栄樹　東京都
ひさびさにメカものだったりする。耳の点はほくろでしょうかピアスでしょうか？　ところで男のピアスと見たがこの顔だとチョット……。

◀佐田　匠　千葉県
こういうネタはけっこう好きですねぇ。体を起こしたときを想像すると，けっこう笑える。これを見ると，ひさびさに遊んでみたくなるね。

本当にいいんですか。どうなってもしりませんよ。二度と戻れないかもしれませんよ。それでもよければ教えてあげましょう。

◆電車でOh!Xを読んでいたとき，ふと目をあげると，向かい側に座っているおじさん（推定年齢56歳）もOh!Xを読んでいるではないか！　私はちょっと感動した。しかし，そんな私に目もくれず，おじさんは次の駅でおりてしまった。

一ノ瀬　宣彦(22)東京都

日常の小さな感動ってやつですね。今度見かけたら声をかけてみては。さらなる感動が待っているかも。

◆ウチの猫が玄関の戸を開けて，外に遊びに行くことを覚えてしまった。まあ，それだけなら良いのですが，家の中の障子は突撃すれば良いことも覚えてしまったようで。お願いだから障子戸も開けて出ていってくれ（涙）。

伊藤　直広(23)福島県

障子に一度板をはって通れないようにしておいたらどうでしょうか？　だめかな。ところで，猫も好きなんだけど，6月くらいに読んだ本の影響で「ゴールデンレトリヴァー」(犬)がとっても飼いたい。名前はやっぱりアインシュタインかな。でもひとり暮らしのアパートじゃ無理か。

◆車をぶつけたショックでこのハガキ出すの忘れてた……ううっ，なんでぶつかっちゃうんだよう……。　　　　美辺　央希(19)東京都

お体，大丈夫ですか？　ところで，事故にあって気を失ったあと，気がついたらなにを最初にしますか？　友人Kは，ぶつけられたあと，ふと気がつくとベッドの上，なぜ自分がここにいるのか記憶がまったくなかった，そこで九九を唱え，「これで頭は大丈夫」と認識したそうな（実話）。

◆近頃，いやなことが多い。しかもそれが現実世界のみならず仮想世界にまで及んできているらしい。何日か前に，牧瀬里穂とひとつコタツで話をする夢を見た。ここぞとばかり愛を語る私に向かって，牧瀬はひとこと，「私，×△さんが好きなの」。なにも夢でまでフラれなくたって……。ちなみに×△とは私の友人だった。

新野　太郎(20)東京都

そんな夢に友人が出てきたってことは，現実世界でその友人とのなにかに敗れたとか，ひょっとして，友人にコンプレックスを持っているとか，まさか友人に恋をしているとか。ウソですよ，冗談，怒らないでね。

◆このハガキが載れば，これ以後Oh!Xを買い続けることだろう。ウソはつきません。ウソつきはキツツキのはじまりといいますから……。

吉岡　昌徳(20)富山県

わーい，これで絶対買い続けてね。でも，“斎藤さん”がカブト虫のように，吉岡さんがキツツキだったらどうしよう。やだー，ちゃんと買ってね。

◆8月号を買った，やっと立体視ができるようになりました。というのか，最近までできなかった理由がわかりました（小学校のときはできていた）。その理由は私の左目の悪さ。なんと，Oh!Xを左目だけで読むには26cm以内に近づかないと見えない。つまり，机に座って背筋を伸ばすとOh!Xさえ見えないのだ。きっと0.00*くらいにはなっているんじゃないかな。だから，今まで私は立体視しようとしても右目だけで見ているようなものなので，できるわけがなかったのだ。だから今では本を机に置いて顔を10cmくらいのところにもってきて立体視してます。早く平行法も身につけたいです。

笹田　泰治(19)愛知県

いやー，私も同じくやっと立体視ができるようになった。これで編集部内で立体視できない人は……。

◆私のタイピングの欠点……それは小指が立っていること。ブラインドタッチなんてゼータクいわんからこのクセをなんとかしたい。

多波見　朋久(20)奈良県

ちなみにこの部分，小指を立てて打ってみた。いつもこんなことしてるんですか。指がつりそう。ひょっとして，そっちのひとじゃないですよね（冗談）。

◆私もブラインドタッチができません。できたらカッコイイかもしれないけど……どうなんだろ。できない理由は，初めてのパソコンがJR-100（おぼえてますか？）だったからかもしれない……ゴムのキーなんてー（いい訳）。

山下　昌俊(23)愛知県

そんなことをいってはいけません。私の近くにはポケットコンピュータから電子手帳，ウォッチコンピュータまでブラインドタッチしてしまう恐ろしい人がいます。

◆1,000円札を出してOh!Xを買い，3歩歩いたときにおつりが100円玉5枚だったのに気づいた。少し両親が悼んだ（字が違った）が，私は歩み出した3歩を無駄にはしなかった。500円のOh!X。ページ数も5/6，なんてことはなかった。それにしても，モラルって大切です。

平山　悟(20)福岡県

このたびはご愁傷さまです。お悔やみ申し上げます。あっ，応対も違った。

◆ある日の深夜，外がさわがしいので起きてベランダに出てみると……なんとマンションの1階が燃えてる。すぐ消防車がきて無事だったけど，いやーびっくりした(僕の部屋は3階)。もう死ぬかと思った。　志摩　憲(19)大阪府

火が消えるまでは，ずっと部屋にいたんですか，それとも，ベランダから消火の様子を眺めていたとか。それにしても火元が1階でよかったね。もし2階か3階だったら……。

◆僕は文学部にいるので男の友達が少なくて困っています。他学部の人にいわせればぜいたくな悩みでしょうけど……。

小山　優一(19)東京都

そうですか。同じ大学の友達じゃないとダメですか？　よかったらお友達になりましょう。なにせこのフロア，女性が2人しかいないものですから。

◆コカコーラは中国語で“可口可楽”（口にすべし楽しむべし）だそうだ。

岸名　紀尚(18)兵庫県

コカコーラとくれば，ペプシコーラはなんて書くんだろう。

◆今，家の外で90cmの水槽で金魚を飼っている。最初20匹だったが，今は5匹しか残っていない。そのかわりに水草は増えるし，でかくなるし，のっとられるのは間近だ。

大橋　修治(25)愛知県

これが載るころには，水槽では水草だけを飼っているのかな。

◆私は小さいころ，砂糖と塩は互いに中和するものだと思ってました。アイスクリームを食べているときに「もっとおいしくしよう」と思い砂糖をふりかけました。そして「パクッ，うげー」。今度は「元に戻そう」と思い，その上から塩をふりかけ「パクッ……」。

村上　学(18)埼玉県

試したことはないけど，小さいころは，僕もそう思っていました。中和しないと気づいたのは，いつのころかな。ミリンの存在を知ってからかな。

◆おお，氏名のところに“フリガナ”とあるではないか。なんでかな？

西山　新志(22)福岡県

それは，みんなの名前が正確に知りたかったからさ。ちゃんと書いてね。

◆Z-MUSICの本がやっと出ますね。ぜひ続刊も出してほしいです。　神野　力(17)愛知県

ところで，お名前なんですけど，本当に“さくらももこ”のマンガのタイトルと同じく「かみのちから」と読むんですか？

◆結婚すると家でコンピュータを触る時間が激減する。会社でも使っている（SEなので）し，夫婦の対話も必要だし，今度既婚者の特集をやってください。　栗本　興一(29)千葉県

たとえば“子育てとX68000”とか，“X68000ですごす夜”てな感じでしょうか。でも，編集部には既婚者が……。

◆「第3次ロボット大戦」に出てくる「サイバスター」ってなにですか？

橋本　誠(19)大阪府

バンプレストのほかのRPGに出てくるオリジナルキャラクタだったかな。サイバスター以外にもバルシオーネとか，あとひとついたけど忘れた。続けてやれば出てくるはずだよ。

◆最近，仕事でCのソースプログラムばかり見ているためか，息抜きで行ったボーリング場のオートスコアの画面から，ソースプログラムが頭に浮かぶ困った私。　小倉　圭司(23)東京都

まだ意識レベルですから大丈夫！　次はきっと夢のなか。

▲玉野　健一　奈良県
ふと，どこかで見かけたような気が，そんな気にさせる女の子。でも，どこで見たんだろう。ちょっと気になったので載せちゃいました。

◆“X68000が欲しい”という人が中学校で5，6人になりました。私のせい？　私って罪な人……。　伊藤　孝(14)神奈川県

周りにX68000ユーザーが増えるといいね。まずはOh!Xをよろしく。

◆ウーロン茶は神葉がいちばんよい。

藤原　利治(26)長野県

個人的にはサントリーのウーロン茶，ビールは……。

◆多重人格ものにハマッてしまって，またノンフィクションものを見つけたのでお知らせします。「ジェニーのなかの400人」というタイトルで，とにかくなんと言っていいのか……できることなら知らないでいたかったような。うまい感想が出てきません。でも，ぜひ読んでほしいと思います。　横山　典俊(23)東京都

読みたいと思う心と読まないほうがという心がせめぎあってます。今度は出版社名を書いていただけるとありがたいんですけど。

◆うわあああ……。ついに僕のタンスが中身をオートイジェクトしてしまった。

木村　弘章(17)茨城県

それは，扉が観音扉のせいね。引き戸の押し入れなら大丈夫！　開けられなくなるかもしれないけど。

◆ううっ！　左耳の下のつけねにシコリが……。蟹じゃなくて，癌だったらどうしよう……。猫にはかまれるし。ノミとりしてたら，ふざけたつもりらしいけれど，まだ子猫なので手加減をしてくれないのであった。狂猫病ってありましたっけ？　小説の中ではあるみたいですけれど……。　枝松　樹(23)愛媛県

蟹でしたらもう少し寒くなったほうがいいんじゃないかと，そうじゃないって。そういえば，このあいだ北海道に行ったとき馬にかまれたあとが腕に残ってる。狂馬病ってあるのかな？　競馬狂病だったりして……。

◆やっぱり，ストIIもいいけど，僕としてはV.R.（バーチャ・レーシング）にも期待している。どっか出してくれないかなあ。きっと僕がX68030を買ったころには出ているにちがいない(勝手な想像)。でも，操作はどうするんでし

◀岩瀬　貴代美　福岡県
卒論のためのフィールドワークでまっ黒に日焼けしているとのこと。イラストの子も日焼けしているみたいで，とってもグーなのね。

◀今井　健生　奈良県
おめでとうございます。XVIをデザインしたあとに彼女ができたということは，やっぱりイメージすることが大切なのかな。どちらも大切にね。

よう？　　　　　　　　半田　将義(19)埼玉県
　バーチャ・レーシングもグーですね。

◆SX-WINDOWS3.0はインストールだけで疲れたが，よさそうな感じ。でも日本中をハイカイしているお化けの9太郎に対抗できるかな。
高村　幸男(61)東京都
　インストール，お疲れさまでした。数のうえでは無理でも質でなら……。

◆ウチのX68000のFDドライブがこわれたせいもあってやっていたバイトが，やっと終わりました。その時のビル（某百貨店の事務所）の下のほうの階を，カタギでない方々が使っていました。出入口のわきのガラスに弾痕が残っていたのでした。そして後日，ニュースを見ているとそのビルが映っている。なんと，某会系の本部だったそうです。　　吉岡　洋明(20)埼玉県
　一歩間違えたら，テレビを見ている側でなく，映ってる側だったかも。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できないこともあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

仲間

★サークル「G.D.T.-夢・旅・人-」では発足にあたり，新規会員を募集します。X68000を中心として，プログラムなどの発表，情報交換などを目的としたディスク会報を中心に，活動していきたいと思っています。興味を持たれた方は，入会案内をお送りしますので，下記宛に62円切手を同封してご連絡ください。〒552　大阪府大阪市港区弁天4丁目13-13　坂田　宗隆

★ディスク会報を発行するサークルを作りますので，会員を募集します。機種は問いません(X68000，X1，MSX，PC-9801，MZ他)。メディアは，MS-DOS(Human68k)系フォーマットで，5インチ2HD，3.5インチ2DD/2HDなどを予定しています(5インチ2D版は考慮中)。会員皆さんからのお便りやQ&A，投稿記事(解析記事，アルゴリズムの解説，バグ情報他)，売買の記事，投稿プログラム，フリーウェアなど（基本的に投稿の種類に制限はありません）を掲載したいと思います。問い合わせは，62円切手を貼った返信用封筒を同封して連絡くださるようお願いします。〒811-42　福岡県遠賀郡岡垣町戸切794-3　筑紫　高宏

★ダイナコン実行委員会です。第11回「ダイナ☆コン」を'93年10月30日（土），31日（日）に開催いたします。内容はSF，特撮，アニメ，パソコンゲームの話題を中心に夜間合宿形式で行います。場所は大名古屋温泉(名古屋市)，参加費用は10,300円です。また，自主企画も同時に募集しています。電子メールでのお問い合わせは，NIFTY-Serve #QGA01713　しんけん2等兵宛です。郵便でのお問い合わせは下記の住所へ62円切手を貼った返信用封筒を同封してください。申し込みの締め切りは，9月30日消印有効です。〒464-91　愛知県名古屋市千種郵便局私書箱90号　ダイナコン実行委員会

売ります

★X1用5インチ2DD「CZ-503F」(1ドライブ)を4,000～5,000円(送料別)で売ります。箱，説明書，IFボード，接続ケーブル，すべてあり。完動。1年ほど使用，その後4年ほど不使用。連絡は希望価格を明記のうえ，往復ハガキでお願いします。〒354　埼玉県富士見市水子6301-14　山下　竜二(23)

★アイ・オー・データ機器製1Mバイト増設RAMボード「PIO-6BE1-A」を7,000円で売ります。X68000ACE/PRO内蔵用です。連絡は往復ハガキでお願いします。〒273　千葉県船橋市市場1-6-9-205　野村　圭一(19)

買います

★X68000 Compact XVI用の2Mバイト増設RAMボード「CZ-6BE2D」を20,000円で，「CZ-6BE2B」とセットなら42,000円で買います。完動品で付属品があればけっこうです。連絡は往復ハガキでお願いします。〒350-13　埼玉県狭山市狭山台2-24-104　木下　卓也(21)

★X68000用増設RAMボード，アイ・オー・データ機器製「PIO-6BE2-2ME」を20,000円，もしくは同社製の「PIO-6BE4-4ME」を35,000円で譲ってください。送料はこちらでもちます。経済力の問題で片方しか買い取れないことをお許しください。連絡は官製ハガキでお願いします。〒106　東京都港区南麻布3-7-13　曹　之祐(19)

★X68000用RGBシステムチューナー(黒)「CZ-6TU-BK」傷なし，箱，マニュアル，付属品つきを，15,000円で譲ってください。名古屋市内，尾張地区の方の場合，車で直接受け取りに行きますが，上記以外の方の場合は，送料をこちらが負担いたします。連絡は，往復ハガキでお願いします。〒509-02　岐阜県可児市鳩吹台7-58　須田　和久(20)

★ネオコンピュータシステムの「POLYPHON」を40,000円＋α(メモリや演算プロセッサの分)，X68000用拡張スロット用2Mバイト増設RAMボードを17,000円前後(メーカーは問わず)で買います。箱はなくてもかまいませんが，付属品などはすべてつけてください。希望価格と＋αの部分を書いて，連絡は往復ハガキでお願いします。〒182　東京都調布市国領町6-20-10　佐藤遊(18)

★X68000 Compact XVI用2Mバイト増設RAMボード「CZ-6BE2D」を20,000円で，「CZ-6BE2B」がセットならば40,000円で買います。どちらも説明書，付属品はすべてつけてください。連絡は往復ハガキでお願いします。〒241　神奈川県横浜市旭区上白根町853　山下　彰(24)

★X68000 XVI用の2Mバイト増設RAMボード「CZ-6BE2A」を20,000円で，「CZ-6BE2B」がセットならば40,000円で買います。連絡は往復ハガキでお願いします。〒306-04　茨城県猿島郡境町1970-1　染谷　達生(18)

★X68000 XVI用の2Mバイト増設RAMボード「CZ-6BE2A」を20,000円で買います。連絡は往復ハガキでお願いします。〒657　兵庫県神戸市灘区篠原台6-28紘心館217　浪越　孝宏(20)

★X68000 XVI用の2Mバイト増設RAMボード「CZ-6BE2A」を20,000円以下で，「CZ-6BE2B」がセットならば40,000円以下で買います。説明書つきのものを，送料込みでなるべく安価でお願いします。連絡は希望価格を書いて，往復ハガキでお願いします。〒939　富山県富山市二俣新町65　野田　康東(17)

★X68000 XVI用の2Mバイト増設RAMボード「CZ-6BE2A」を25,000円で，「CZ-6BE2B」がセットなら55,000円で，さらに「CZ-6BE2B」が2個セットなら85,000円で買います。バラなら1個につき25,000円です。箱はなくてもかまいませんが，説明書はつけてください。送料はこちらが負担します。連絡は往復ハガキでお願いします。〒961　福島県白河市女石15-7　斎藤　洋(20)

★HAL研究所のハンディスキャナ「HGS-68」を15,000円で買います。付属品，説明書をつけてください。連絡は往復ハガキでお願いします。〒262　千葉県千葉市花見川区長作町515-2　小沢　一生(22)

★MZ-2500増設RAMボード「MZ-1R26」を17,000円前後，辞書ROMボード「MZ-1R28」を11,000円前後で買います。送料はこちらが負担します。連絡は往復ハガキでお願いします。〒569　大阪府高槻市登町14番A18-202　川端　行雄(56)

★X68000用SCSIボード「CZ-6BS1」を12,000円で，箱なし可，付属品，説明書は必要。まずは，往復ハガキで連絡をお願いします。〒228　神奈川県相模原市鵜野森グリーンハイツD9-101　浅井　徹(22)

★X68000 XVI用の2Mバイト増設RAMボード「CZ-6BE2A」を送料込み20,000円で買います。連絡は往復ハガキで。〒737　広島県呉市弥生町6-33　谷本　和生(40)

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今月は8月号の内容に関するレポートです。

●8月号の特集では,「MAKEを使おう」がよかった。実際に使ってみるまでMAKEがこんなにも便利だとは思いませんでした。もともとC言語は使えないし，アセンブリ言語で書いた100行程度のプログラムひとつアセンブルするだけの私には，まったく必要ないと思っていました。そのため「MAKE?　なにそれ」といった状態でした。しかし，この記事を読んで初めてそのよさを知りました。私もそろそろC言語を勉強しようと思っていますので，いまのうちからMAKEファイルを習慣づけておこうと思います。

森崎　剛(21)　X68000 XVI　広島県

●8月号の特集を読んで「はてさて，これほどまでにその存在を切望された言語だったか？」なんて考えてしまいました。普及の第一の理由は,「安価なコンパイラが多数出回ったから」と思ってしまいます。特集にもあるようにあれほど初心者はおろか，上級者（素人に教えるときなど）まで泣かせる言語。そうそうはやるものじゃないでしょう。難しいといわれながらも，ここまで市民権を得たのはなぜなのでしょう？　特集を読むにつれて疑問ばかりが出てきました。私にとってC言語は，ミステリーだったようです。ただいえることは,「ほかの高級言語にないアナーキーな仕様が私好みだ」です。

中矢　史朗(22)　X68000 ACE-HD,X68030,PC-386P　愛媛県

●特集の「C言語をめぐる状況」はいいですよね。XCを買っておきながら，C言語をちっとも活用していない私にたくさんのいいわけを用意してくれました。記事の中でもいくつかC言語のもつ取っつきにくさが挙げられていますが，私の場合は，GCCをインストールできなかったのが敗因です。現在は，一応GCCも動いていますが，出鼻をくじかれたのは痛かったです。これからC言語を始めようとしている人は，XCとGCCのどちらを選ぶのでしょうか。GCCは現在でもその気になれば通信で入手できます。XCはいまでも高価ですが，グラフィック，音源，ジョイスティックなどを手軽に扱えるのは魅力です。私も他人ごとではありませんが，C言語を学ぼうとしている人はがんばってください。

中村　健(23)　X68000　ACE-HD,PC-386GS,AMIGA500　埼玉県

●ここのところ「夏真っ盛り，アマチュアリズムのX68000」のような，妙に人なつっこい特別企画が多くて好きです。今回は,「ある電脳絵師のひとり言」がとてもよかったです。私が我流のCGを描き始めて1年，ほかの人の作品を見るたびに「どうやって描くのだろう」と思っていました。川原氏の記事は，CGの制作過程がわかり，それもカラーの記事だったので参考になりました。ただ，もう少し内容を充実してほしいな，とも思います。かぎりある誌面，ほかにも音楽や通信の記事もありますからしかたのないことかもしれません。個人的に希望をいわせてもらえれば，もう少しいろいろな人にCGの描き方についての記事(手本)を描いてもらいたいし，ページ数も「CGの特集」といえるくらいのボリュームにしてほしいと思いました。

野原　賢次(32)　X68000 ACE-HD,X1,X1turbo model30　埼玉県

●8月号をパラパラめくると，なにやら体格のよいお姉さんが寒そうな格好で……「？8月号の特集はC言語だったはずだが。ま，いいか。暑いしね。ははは……」と長袖のシャツ姿でとりつくろう。なにも悪いことをしていないのに慌てて次のページへ逃げる。するとどうだろう。今度は2ページにわたってお姉さんが散らばっているではないか……。ま，お姉さんはともかく，特別企画にあった伊藤氏の「タッチタイピングへの野望」の記事で書かれていた「同時進行プログラミング」ともいうべき展開は，私も好きです。しかし，こういう文章は，初心者がBASICを知るのに向いていそうで向いていないな，とも思いました。なぜなら，初心者は「配列に入れる」というような表現が，最もわからないからです。伊藤氏の記事では，そういう言葉のあと，すぐに実践されていてよかったですが。Oh!Xにはもっとそういう配慮がほしいですね。

林　大助(17)　X68000 SUPER,PC-8801mkIIFR　神奈川県

●新製品紹介にあった「Easydraw SX-68K」は前々からほしかったので，バンザーイと喜んでいたらシャーペンとの間でカット＆ペーストができないとのこと。ガーン。記事ではクリップボードのせいなのか，シャーペンのせいなのかよくわからなかったのですが，とりあえずシャープにはなんでも取り込むクリップボードとなんでも貼りつけられるリッチテキスト形式を望みましょう。「Easydraw SX-68K」本体はよくできているみたいだけに残念です。特別企画は「アマチュアリズムのX68000」なのにX1turboの記事があったりしてナイスですね。やっぱりものが作られていく過程は面白い。川原さんはいつもこんなして絵を描いているのか。こんなマネはとてもでき～ん。もう少し記事を増やしてこっちを特集にしてしまえばよかったのに。

石田　伯仁(20)　X68030,MZ-731,PC-8801mkII MR,PC-E200　神奈川県

●毎回楽しませてもらっている「知能機械概論」ですが，8月号は「研究室という環境」という，身近な話という点で興味深いものがありました。教官室のレイアウトをああいうふうに分類されると，なにか新鮮な感じがします。私のゼミの教授の教官室は図4タイプですが，個人的には図5タイプが出入りしやすそうですね。なにかぜんぜんOh!Xと関係ない話ですけど。

北風　保(21)　X68000 ACE　東京都

ごめんなさいのコーナー

9月号　ペンギン情報コーナー

P.139　新製品情報として紹介した，X68000/030用98バスマウスアダプタ「MK-MJ1」の定価が3,500円(税別)となっていましたが，都合により4,000円(税別)に変更されました。価格変更に伴いご迷惑をおかけしましたことをおわびします。

バグに関するお問い合わせは
☎03(5642)8182(直通)
月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

夢は見るもの 野望は つかむもの

▶１年以上のご無沙汰，特別付録「秋祭りPRO-68K」は，皆さんの期待を裏切らない内容でしたでしょうか。

いつもどおり詰め込むだけ詰め込んだ結果，1.4Mバイトフォーマットが必要になり，解凍後はディスク６枚組になってしまう豪華版。解凍作業はバッチファイル，かろうじてシステムを入れることができたぐらいですから，密度は保証つきです。

付録ディスクの目玉「SLASH ver.1.0」やSX-WINDOWのツール，そしてゲームなど，使えるものから遊べるものまで，いろいろな作品を用意しました。開発系のアプリケーションが多いのは，Oh!Xならではの付録ディスクといえるでしょう。

解凍してただ遊ぶだけの楽しい付録ディスクもいいでしょうが，読者の皆さんには，ぜひ自分の作りたいものを作る喜びを感じてもらいたいものです。できないとあきらめずにできる限り挑戦してみてください。

▶また，今回は，評価版というものも多少含まれています。特に「SLASH」は，まだ先を見切れていない部分もあって仕様が確定していません。そのため，「SLASH」開発ツールもあまり完成度の高いものではありません。それでも掲載に踏み切ったのは，読者の皆さんにぜひともこの「SLASH」の素晴らしい世界を感じ，協力してもらいたかったからです。

横内氏は，たったひとりの力でシステムを作り上げました。Oh!Xの読者が集まれば，なにかもっとすごいことができそうだと思いませんか。そうして，皆さんの協力によって「SLASH」の世界は，これからも進化していくことでしょう。

また，今月号から始まった連載「ハードコア3Dエクスタシー」とともに，横内氏&丹氏のパワーに期待してみようではありませんか。具体的な意見，要望，疑問なども随時受けつけています。どしどしお寄せください。

▶最後に，10月号の定価がやむなく800円となってしまいましたことをおわびします。

投稿応募要領

●原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。

●プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。

●ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。

●投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒103 東京都中央区日本橋浜町3-42-3

ソフトバンク出版部

Oh!X「㋢㋮㋱」係

SHIFT・BREAK

▶コンビニではガム１個でも袋に入れてくれるが，ボクは「テープでいいです」ということにしている。環境問題もあるが，すぐに捨てるとわかっているものをもらうのはムダだからだ。そんな僕がいちばんハラが立つムダは3.5インチディスクの１枚１枚についてくるプラスチックケース。こんなものをなくしてその分値段を下げてほしいと思わない？(E.K.)

▶社会人になった。毎朝だいたい同じ時間に起きて，だいたい同じ時間に食事をとって，だいたい同じ時間に家を出て，だいたい同じ時間に会社に着く。規則正しい生活で体重も増えた。環境の変化にも慣れた今では，変化のない毎日を暮らしている自分がマジに嫌になってきた。自分の本当にやりたいことをやろうと真剣に思うこの頃。(欲求不満のASA)

▶今月はえらくハード。でもこのあと引っ越しして学校行ってまたハード。東京ともしばらくサラバだ。さて，念願のデルソルを購入。キサマも地に落ちたな，とあるスピード狂から攻撃を受けた。資金が足りないよ。でもいずれファミリーカーとしてNSXを……ナマいってると次はロータス買って泣かすぜ。戦車かアパッチでも可。(ビッグマウス威)

▶悩んだあげく，「完全自殺マニュアル」という本を購入。死ぬほど，その方法について解説してあるので，一気に読んでしまった。一番バカバカしいと思ったのは，塩でやるというやつで，300g程度飲み込むと，ヤバイらしい。いかにもな薬がヤバイのは理解できるが，塩で死ねるとは信じ難い。ぜひ，体験者の手記を見てみたいものだ，うんうん。(八)

▶ツーリングで高知に行った。道はいいし信号は少ないし，帰ってきたらエンジンの調子も明らかによくなっていた。やっぱり東京は単車乗りの環境としては病んでいるのだろう。あまり観光はしない主義だが，高知城にだけは行った。夜の高知城はライトアップされていて妙に感動。ちょいとミーハーな行動だったけど，行ってよかった，うん。(A.T.)

▶新宿駅。JR中央線総武線都営新宿線営団丸の内線JR山手線。台風。都心から新宿駅へ向かう路線が地下鉄都営新宿線以外みなストップ。さあ大変。新宿から郊外へ出ようとする人々が，一斉に都営新宿線へ殺到したのである。そんなん，さばききれるわけないのだ。都心交通網の脆弱さは，常にキャパシティギリギリで運営されていることだと実感。(K)

▶突然水道が止まった。風呂に入れないのはなんとか我慢できる。歯磨きできないのはちょっと嫌だ。しかし，トイレが使えないのは致命的だった。さんざんあおったのだが，その日の内に修理はこなかった。しびれを切らしてビジネスホテルに泊まることを決意。何軒か電話をかけて見つけたホテルは少し怪しい。１人でも２人でも同じ料金とは。(KO)

▶これが出るころは，筋肉痛で苦しんでいるだろう。長野県の山奥へ，MTBの大会に行ってきやす。でも，どうかしてるよなあ。４時間でコース何周できるかだって。ここのところMTBなんて乗ってる時間なかったのに。もちろん理由は……。異動から１カ月，皆さんに迷惑をかけつつ，少しずつ慣れてきやした。遅ればせながら，これからよろしく！(高)

▶平和主義者のあたしは太陽が眩しくたって，人殺しなどいたしません。だから夏はやっぱり暑いほうがいい。傘さして歩くのもうっとうしいから会社に閉じこもる。あげくの果てに台風襲来で電車が不通，お家に帰れなくなる。運動不足解消には会社で「あばれる」あ〜あ。せめて秋には旅行でも……。式根島に行ったのが今年唯一の夏でした。(ふ)

▶８月半ばに誕生日を迎え，せっかくの誕生日だから，ということで出不精の僕が日本海を見に新潟まで出かけた。結局，天候に恵まれず，しかも２日目は怒涛の雨攻撃。しかたないので，新潟駅付近のパチンコ屋巡りで時間を潰すハメに……結果は，プリンセスに２万円貢いでボロ負け。それにしても隣の奴の保留玉４連チャンは目の毒だった。(J)

▶付録ディスクをつけた直後というのは，どうしてももう二度とやんねーぞときう気持ちになるらしい。今回は生命の危機は感じずにすんだが，1.4Mバイトは凄く面倒ということがわかった。しかし横内君の作るプログラムはよく縮む。収録はされてないが，LHAで０％になる実行ファイルなんて初めて見たぞ。Z-MUSICは遅れている。ごめんなさい。(U)

▶台風の朝，近所のコンビニへ。買物カゴをレジに乗せたところで，財布を忘れたことに気がついた。「すみません」と慌てて財布を取りに帰り，再びコンビニに戻ったが，なんと財布はほとんどカラ状態。涙の３往復で身も心もズブ濡れとなってしまった。今月は，Oh!Xを買いにいったが，お金が足りなかったなんて人がいなければいいんだけど。(T)

microOdyssey

文化史の本によると，ルネサンス期に考えられていた人間の理想像は「あらゆるものについてひととおりの知識を持ち，なおかつそのうちの何点かについては精通している」というものだそうだ。オールマイティかつスペシャリストという，ちょっととんでもない話だが，「あらゆるもの」「精通」はあくまで理想だから到達できるかどうかは別にして，そういう志向性を持っているのは悪いことではないだろう。

かの時代の天才たちの筆頭に挙げられるのがレオナルド・ダ・ヴィンチだ。有名なのは「モナ=リザ」だが，彼は画家であるだけではなく建築，天文学，解剖学，音楽……といろいろなことをしていて，さらにそのどれもが一流，というスーパースペシャルなひとだったという。

私が不思議に思ったのは，能力はさることながら，そんな時間がどこにあったのかということだった。時間は物理的に限られているし，人間には生きるために必要な時間もある。どうしてそんなにいろいろなことができたのだろうか。学問というのは相互に関連があるから，ほかの分野の知識があるぶん，新しいことの理解や吸収は効率がいいだろうとしても，である。

この疑問はしばらくのあいだ私のなかでそのままになっていたが，その後ちょっとだけ解けた。彼は極端なひと嫌いで，友人も作らず，ひたすら孤独にものごとに熱中していたのだそうだ。だから人づきあいで煩わされることも女性問題で悩むこともなく，ひたすら自分の興味のおもむくままに学問に没頭していたらしい。それが幸福かどうかは異議を唱えるひともいるだろうが，やりたいことをして過ごしていたのなら，幸福な人生だったのだろうと私は思う。

しかし，いずれにしても現代ではそういう隠遁生活は学問をするうえで効率よいやり方ではないだろう。世界が拡がり，それぞれの分野の研究は深まり，変化のスピードは加速している。なによりも，情報の種類や収集の方法が増え，それらの活用次第で結果は全然違ってくる。

私は，ダヴィンチ生活への覚悟を決めるほどには俗世間の誘惑に勝てないし，それはまあ，しかたがない。でも，ルネサンス的理想もまだ捨ててはいない。ダヴィンチになれるとは思わないが，それはあきらめとか挫折ではない。私にとって重要なのは，どこまで到達できたかということよりも，あらゆる方向を見ようとする意志をもち続けることのほうだ。

いまの時代に生きている私たちにとっては，隠遁生活にはいるよりも，効率よい情報伝達を考えることのほうが有効なのだろう。伝達手段は増えたが，それを選択したり使ったりすることにあまりにも時間をとられてしまったら，それは本末転倒になってしまう。やりたいことを実現するためにパソコンを使うのに，システム設定などに時間がかかり過ぎる，みたいな問題だ。そんなことはさっさとクリアして，本来の「やりたいこと」に時間を使いたい。

ところで，ダヴィンチは左利きで，彼の書く文字はすべて鏡文字になっている。読むものと書くものが裏返しの関係で，つまり脳を中心としてINとOUTが違っているわけだ。このことも，私にはなんとなく興味深い。もしかしたら，これは彼のものごとの認識に影響を及ぼしていたのではないだろうか。　(ふ)

1993年11月号10月18日(月)発売

特集　SLASHの活用
- ポリゴンシステムの可能性
- 簡易回転体モデラ

ちょっと怪しいCASLOAD/CASSAVE

新製品紹介
Easydraw SX-68K/MATIER ver.2.0

全機種共通システム
基礎から学ぶアセンブラ入門(仮題)

バックナンバー常備店

東京	神保町	三省堂神田本店5F 03(3233)3312
	〃	書泉ブックマートB1 03(3294)0011
	〃	書泉グランデ5F 03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン 03(3257)2660
	八重洲	八重洲ブックセンター3F 03(3281)1811
	新宿	紀伊国屋書店本店 03(3354)0131
	高田馬場	未来堂書店 03(3209)0656
	渋谷	大盛堂書店 03(3463)0511
	池袋	旭屋書店池袋店 03(3986)0311
	八王子	くまざわ書店八王子本店 0426(25)1201
神奈川	厚木	有隣堂厚木店 0462(23)4111
	平塚	文教堂四の宮店 0463(54)2880
千葉	柏	新星堂カルチェ5 0471(64)8551
	船橋	リブロ船橋店 0474(25)0111
	〃	芳林堂書店津田沼店 0474(78)3737
	千葉	多田屋千葉セントラルプラザ店 043(224)1333
埼玉	川越	黒田書店 0492(25)3138
	川口	岩渕書店 0482(52)2190
茨城	水戸	川又書店駅前店 0292(31)0102
大阪	北区	旭屋書店本店 06(313)1191
	都島区	駸々堂京橋店 06(353)2413
京都	中京区	オーム社書店 075(221)0280
愛知	名古屋	三省堂名古屋店 052(562)0077
	〃	パソコンΣ上前津店 052(251)8334
	刈谷	三洋堂書店刈谷店 0566(24)1134
長野	飯田	平安堂飯田店 0265(24)4545
北海道	室蘭	室蘭工業大学生協 0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある『新規』『継続』のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

基本的に，定期購読に関することは販売局で一括して行っています。住所変更など問題が生じた場合は，Oh!X編集部ではなくソフトバンク販売局へお問い合わせください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS(株)にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(3238)0700

Oh!X 10月号
■1993年10月1日発行　定価800円(本体777円)
■発行人　橋本五郎
■編集人　稲葉俊夫
■発売元　ソフトバンク株式会社
■出版事業部　〒103 東京都中央区日本橋浜町3-42-3
Oh!X編集部 ☎03(5642)8122
販売局 ☎03(5642)8100 FAX 03(5641)3424
広告局 ☎03(5642)8111
■印刷　凸版印刷株式会社

満開の電子ちゃん

作：いわいいっぺい
え：岡村祭

講読方法：定期購読もしくはソフトベンダーTAKERUでお買い求めいただけます。

★定期購読の場合＝購読料６ヶ月分6,000円（送料サービス、消費税込）を、現金書留または郵便振替で下記の宛先へお送り下さい。

現金書留の場合：〒171 東京都豊島区長崎1-28-23 Muse西池袋2F ㈱満開製作所
郵便振替の場合：東京 5－362847 ㈱満開製作所

- ご注文の際は、郵便番号・住所・氏名・電話番号を忘れずに記入して下さい。
- 3.5インチディスク版をご希望の方は、「3.5インチ版」とご指定下さい。
- 新規購読の方は「新規」と明記して下さい。なお、特に購読開始号のご指定がない場合は既刊の最新号からお送りいたします。
- 製品の性格上返品には応じられませんが、お申し出があれば定期購読を解約し残金をお返しします。

★TAKERUでお求めの場合＝１部につき1,200円（消費税込）です。

- 定期購読版と内容が一部異なる場合があります。御了承下さい。
- お問い合わせ先 TEL(03)3554－9282（月～金 午前11時～午後６時）

（なお、定期購読版のバックナンバーについては定期購読の方のみご注文を承ります）

船越 直弥（北海道）

前回の私の推薦文が掲載されてすぐ、後輩からX68Kを購入しました。そこで電脳倶楽部を定期購読するべきか迷っています。こんな怪しいディスクマガジンを定期購読して良いのだろうか？ 今のところ、タケルでときどき電倶を購入しているのですがどうしても定期購読する勇気がないのです。しかし、このままでは祝様に面目がたちません。こんな私はどうすれば良いのでしょうか。もう少しタケルで様子を見たいと思います。

ということで、合言葉は「電源オンですぐ起動、マウスひとつでらくらく操作」ですね。

お支払いは、便利な商品到着払い(手数料要)をご利用下さい。

《業界No.1の"P&Aメンテナンスサポート"》

最高の保証システム

①業界最長の新品パソコン5年保証
（※モニター・プリンター3年間保証!!※一部商品は除きます。）
②中古パソコンの1年間保証
（モニター・プリンター6ヶ月間保証）
③初期不良交換期間3ヶ月
（※新品商品に限らせていただきます）
④永久買取保証
⑤配達の指定OK!!（土曜・日曜・祭日もOK!!）
⑥夜間配送もOK!!
（※PM6:00〜PM8:00の間　※一部地域は除きます。）

便利でお得な支払いシステム

①翌月一括払い手数料無料（ご利用下さい。）
②業界No.1の低金利
③月々の支払いは￥1,000より
④9ヶ月先からのスキップ払いOK!!
⑤84回までの分割、ボーナス併用OK!!
⑥カレッジクレジット
⑦ステップアップクレジット
⑧ボーナスだけで10回払いOK!!
⑨現金一括払いOK!!
⑩商品到着払いOK!!（代引き手数料が必要になります。）
（※商品・金額ご確認の上、銀行振込・現金書留にてご入金下さい。）

●法人向けリースシステム
業務に最適なシステムを構築します。損金処理が可能なリース契約をどうぞ。

SHARP＝X68030エキスパートショップ

P&A＝X68030 ダブルNEWフェア
《9月18日〜10月17日》

注目!! 冬のボーナス一括払い手数料(金利)無料（平成5年10月末/11月末/12月末のいずれかをご指定ください。）

32ビットX68030いよいよ登場。購入ダブルチャンス!!

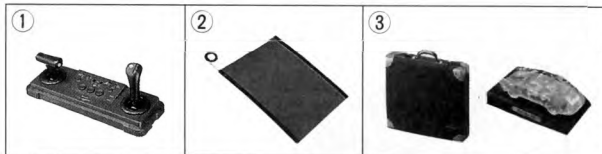
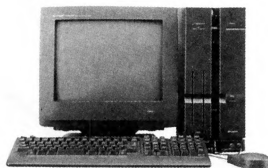
X68030発売記念

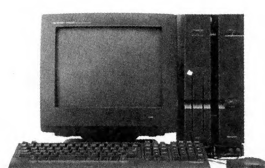
X68030をモニターとセットで／単品で　購入の方!!

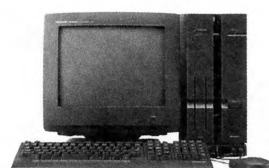
さらに現在お持ちのパソコンと下取り交換されたお客様に期間中もれなく、

①サイバーステック……………………（CZ-8NJ2 ￥23,800）
②CRTフィルター……………………（BF-68PRO ￥19,800）
③X-68000フロッピーアタッシュケース（￥8,000）
とクリスタルポルシェ（￥8,000）

以上のいずれかプレゼント!!

①

②
③
今だからこそ選ぶ ズバリお買い得セット

①
通信セット
- CZ-500C＋CZ-608D
- MC-14400FX（FAXモデム、マイクロコア）
- CZ-257CSD（communication）

合計定価￥559,400
P&A超特価 **￥396,000**

②
ハードディスクセット
- CZ-500C＋CZ-608D
- LHD-FM100E（ロジテック、100MB）
- ケーブル付

合計定価￥598,600
P&A超特価 **￥396,800**

③
MIDIミュージックセット
- CZ-500C＋CZ-608D
- SX-68MII（システムサコム）
- CM-300（ローランド）

合計定価￥570,600
P&A超特価 **￥406,000**

※本広告の掲載の商品の価格については、消費税は含まれておりません。

P&A 全国通販

★頭金なし!!
★即日発送!!

32ビットX68030いよいよ登場（送料￥2,000・消費税別）

④ グラフィックセット

- CZ-500C＋CZ-608D
- HS-7RII（オムロン、スキャナ）
- Z's STAFF PRO-68K Ver.3.0（ツァイト）

合計定価￥590,600
P&A超特価
￥411,000

⑤ 光磁気ディスクセット

- CZ-500C＋CZ-608D
- CS-M120（コパル、光磁気）
- ターミネータ、ケーブル付
- MOカートリッジ（1枚）

合計定価￥670,800
P&A超特価
￥477,000

本体の変更

①CZ-510Cに変更の場合￥71,000
②CZ-300Cに変更の場合￥ 1,000
③CZ-310Cに変更の場合￥64,000
加算して下さい。

モニターの変更

①CZ-607D（チューナー付）に変更の場合￥ 3,000
②CZ-614D（チューナー付）に変更の場合￥31,000
③CU-21MD に変更の場合￥60,000
加算して下さい。
※300シリーズにチューナー付のモニターを接続の場合CRTケーブルを購入して下さい。

※お支払いは、便利な商品到着払い（手数料要）をご利用下さい。

旧シリーズ今が買いどき!!（クレジット表：送料、消費税込み）
X68000 Compact XVI/XVI 送料￥2,000、消費税別

Compact XVI

①
- CZ-674C-H（本体）
- CZ-608D-H（モニター）

定価￥392,800
P&A超特価**￥173,000**

12回	15,800	24回	8,400	36回	5,800	48回	4,500	60回	3,800

XVI

①
- CZ-634C-TN（本体）
- CZ-608D-H（モニター）

定価￥462,800
P&A超特価**￥213,000**

12回	19,500	24回	10,300	36回	7,100	48回	5,600	60回	4,700

上記のモニターをCZ-614Dに変更

②
- CZ-674C-H（本体）
- CZ-614D-TN（モニター）
- CZ-6CR1（RGBケーブル）
- CZ-6CT1（TVコントロール）

定価￥443,000
P&A超特価**￥213,000**

12回	19,500	24回	10,300	36回	7,100	48回	5,600	60回	4,700

上記のモニターをCZ-614Dに変更

②
- CZ-634C-TN（本体）
- CZ-614D-TN（モニター）

定価￥503,000
P&A超特価**￥243,000**

12回	22,200	24回	11,700	36回	8,100	48回	6,400	60回	5,300

※上記①のモニターをCZ-607D-TN（定価￥99,800）に変更の場合 **￥3,000** 加算して下さい。
※ディスケット10枚プレゼント。

カラーイメージスキャナー

⊙JX-220X 限定10台
（RS-232Cケーブル・ユーティリティソフト付）
定価￥168,000▶特価**￥99,800**

X68000/68030専用ハードディスク

外付

■富士通
⊙FMHD-1201G（120MB、17ms）
…………定価￥70,000▶特価**￥49,800**
⊙HD-K200A（モッキンバード）（200MB、13ms）
…………定価￥79,800▶特価**￥61,000**

■ロジテック
⊙SHD-FMX120（120MB）
………定価￥ 59,800▶特価**￥43,500**
⊙LHD-FM200E（200MB、17ms）
………定価￥138,000▶特価**￥63,000**

■ジェフ
⊙GF-200（200MB、15ms、64K）‥定価￥ 98,000▶特価**￥ 59,000**
⊙GF-240e（240MB、15ms、64K）‥定価￥118,000▶特価**￥ 63,500**
⊙GF-340i（340MB、14ms、64K）‥定価￥158,000▶特価**￥ 87,800**
⊙GF-540i（540MB、8.5ms、256K）‥定価￥238,000▶特価**￥151,800**

内蔵

■CZ-500C/300C専用
⊙CZ-5H08（80MB、23ms）
………定価￥ 98,000▶特価**￥71,800**
⊙CZ-5H16（160MB、18ms）
………定価￥135,000▶特価**￥99,500**

X68000シリーズ～P&Aスペシャルセット（送料￥2,000・消費税別）

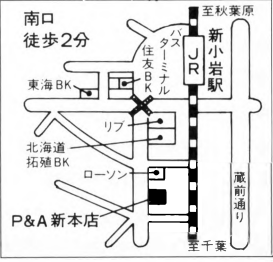
SUPER-HD ★ハードディスク81MB搭載!! ※ディスケット10枚・ゲームソフト1ヶプレゼント

Ⓐセット：■CZ-623C-TN（単品） …………定価￥498,000▶特価￥158,000
Ⓑセット：■CZ-623C-TN＋CZ-606D ……定価￥577,800▶特価￥213,000
Ⓒセット：■CZ-623C-TN＋CZ-608D ……定価￥592,800▶特価￥226,000
Ⓓセット：■CZ-623C-TN＋CZ-607D ……定価￥597,800▶特価￥228,000
Ⓔセット：■CZ-623C-TN＋CZ-614D ……定価￥633,000▶特価￥248,000
Ⓕセット：■CZ-623C-TN＋CU-21HD ……定価￥646,000▶特価￥258,000

〔銀行振込でお申し込みの方〕（電信扱いでお振込み下さい。）
〔振込先〕さくら銀行 新小岩支店
当座預金 2408626
（株）ピー・アンド・エー

超低金利クレジット率

回数	3	6	10	12	15
手数料	2.9	3.9	4.9	5.4	8.4
回数	24	36	48	60	72
手数料	11.4	15.9	20.9	26.9	34.9

P&A 株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目2番地20号
☎03-3651-0148（代） FAX.03-3651-0141
●営業時間：AM10:00～PM7:00
日・祭：AM10:00～PM6:00
●定休日／毎週水曜日

●価格は流通事情により変動致しますので、銀行振込・書留等の送付前にあらかじめお電話にてご確認下さい。

※お支払いは、便利な商品到着払い（手数料要）をご利用下さい。

P&Aならではの 新品パソコン 5年保証

《業界No.1の"P&Aメンテナンスサポート》

最高の保証システム
①業界最長の新品パソコン5年保証
（※モニター・プリンター3年間保証!! ※一部商品は除きます。）
②中古パソコンの1年間保証
（モニター・プリンター6ヶ月間保証）
③初期不良交換期間3ヶ月
（※新品商品に限らせていただきます）
④永久買取保証
⑤配達の指定OK!!（土曜・日曜・祭日もOK!!）
⑥夜間配送もOK!!
（※PM6:00〜PM8:00の間　※一部地域は除きます。）

便利でお得な支払いシステム
①翌月一括払い手数料無料（ご利用下さい。）
②業界No.1の低金利
③月々の支払いは¥1,000より
④9ヶ月先からのスキップ払いOK!!
⑤84回までの分割、ボーナス併用OK!!
⑥カレッジクレジット
⑦ステップアップクレジット
⑧ボーナスだけで10回払いOK!!
⑨現金一括払いOK!!
⑩商品到着払いOK!!（代引き手数料が必要になります。）

（※商品・金額ご確認の上、銀行振込・現金書留にてご入金下さい。）

モデム（送料¥1,000）
マイクロコア◆MC-14400FX（定価¥46,800）特価¥34,500
富士通◆FMMD-3111G（定価¥35,800）特価¥24,800
オムロン◆MD-24XT10V（定価¥29,800）特価¥22,800
◆MD-96XT10V（定価¥46,800）特価¥35,500
アイワ◆PV-AF144V5（定価¥64,800）特価¥49,000

●お近くの方は、お立寄下さい。専門係員が説明いたします。
●本体単品でも受付します。詳しくは、お電話にてお問合せ下さい。

周辺機器コーナー （送料¥1,000・消費税別）

No.	品名	定価	特価
1	BF-68PRO	¥19,800	¥14,400
2	CZ-8NM3	¥9,800	¥7,200
3	CZ-8NT1	¥13,800	¥10,000
4	CZ-6BE2A	¥59,800	¥42,800
5	CZ-6BE2B	¥54,800	¥39,300
6	CZ-6BE2D	¥54,800	¥39,300
7	CZ-6BF1	¥49,800	¥35,800
8	CZ-6BP1	¥79,800	¥57,000
9	CZ-6BM1	¥26,800	¥19,300
10	AN-S100	¥36,600	¥26,300
11	CZ-6SD1	¥44,800	¥32,500
12	CZ-6BN1	¥29,800	¥21,500
13	CZ-6BV1	¥21,000	¥15,200
14	CZ-6BC1	¥79,800	¥57,000
15	CZ-6BG1	¥59,800	¥43,000
16	CZ-6BU1	¥39,800	¥28,500
17	CZ-6PV1	¥198,000	¥142,000
18	CZ-6BS1	¥29,800	¥21,500
19	CZ-8NJ2	¥23,800	¥17,500
20	CZ-6BL2	¥298,000	¥214,000
21	CZ-6CSI(674C用)	¥12,000	¥8,900
22	CZ-68HA		¥91,000
23	CZ-6CR1(RGBケーブル)	¥4,500	¥3,600
24	CZ6CT1(テレビコントロール)	¥5,500	¥4,400
25	CZ-6BP2	¥45,800	¥33,300

■システムサコムボード
●SX-68MII（MIDI）定価¥19,800▶特価¥13,500
●SX-68SC（SCSI）定価¥26,800▶特価¥17,500

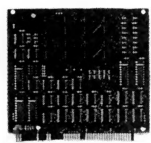
周辺機器特選品コーナー

JX-325X カラーイメージスキャナ　定価¥190,000　特価¥143,000
JX-32F12（写真上部分）定価¥148,000　特価¥112,000
CZ-6VTI カラーイメージユニット　定価¥69,800　特価¥49,500
CZ-6TU RGBシステムチューナー　定価¥33,100　特価¥23,900
JX-220X カラーイメージスキャナ《限定》　定価¥168,000　特価¥99,800
CZ-8NSI カラーイメージスキャナ　定価¥188,000　特価¥133,000
（X68030用）増設RAMボード＆数値演算プロセッサ
CZ-5BE4 定価¥54,800　特価¥42,000
CZ-5ME4 定価¥49,800　特価¥38,000
CZ-5MPI 定価¥54,800　特価¥42,000

FDD（5インチ×2基）
■CZ-6FD5（シャープ）（定価¥99,800）
P&A超特価 ¥49,800

プリンター（ケーブル用紙付・送料¥1,000・消費税別）
■CZ-8PC5-BK 定価¥96,800 特価¥68,500
■CZ-8PK10 定価¥97,800 特価¥71,000

カラーイメージジェット
■IO-735X-B 定価¥248,000
特価¥135,000
（送料・消費税込み¥140,080）

X68030/68000 メモリボード（I/Oデータ）

①SH-5BE4-8M（X68030用）（送料・消費税込み¥47,586）特価¥45,500
②SH-6BE1-1ME（600C専用）（送料・消費税込み¥12,669）特価¥11,600
③1MB増設RAMボード（ACE/PRO/PROII用）（送料・消費税込み¥12,669）特価¥11,600
④2MB増設RAMボード（拡張スロット用）（送料・消費税込み¥24,411）特価¥23,000
⑤4MB増設RAMボード（拡張スロット用）（送料・消費税込み¥40,170）特価¥38,300

P&A特選パソコンラック＆OAチェアー （消費税込み）（送料無料、離島を除く）

①3段 ¥8,240　②4段 ¥9,785　③5段 ¥11,845
※全機種→キャスター付
※上から2番目棚板移動可能（4/5段）
※フレーム色：4段→黒、3/5段→ホワイト

①¥9,270 ●布張り（ダークグレー）●ガスシリンダー
②¥13,390 ●布張り（ダークグレー）●ガスシリンダー●肘付

X68000用ソフトコーナー （送料¥700・消費税別）

品名	定価	特価
◆Z'sSTAFFPRO68KVer.3.0（ツァイト）	¥58,000	¥37,500
◆Z'sTRIPHONYデジタルクラフト（ツァイト）	¥39,800	¥27,000
◆テラッツォ（ハミングバード）	¥19,400	¥13,600
◆ラジックパレット（ミュージカルプラン）	¥19,800	¥14,200
◆たーみのる2（SPS）	¥17,800	¥13,000
◆Mu-1Super（サンワード）	¥39,800	¥28,500
◆CMA68K（シティソフト）	¥29,000	¥21,800
◆サイクロンEXPRESSα68	¥98,000	¥69,000
◆C-TRACE68Ver.3.0（キャスト）	¥98,000	¥68,500
◆OS-9/X68030 V.2.4.5（マイクロウェアシステムズ）	¥25,000	¥19,900
◆C&ProfessionalPackV3.2（マイクロウェアジャパン）	¥80,000	¥57,800
◆ウエットペイント1〜3（ウエーブトレイン）〔各〕	¥15,000	¥11,500
◆マチエール（サンワード）	¥39,800	¥28,800
◆WindexPRO68（JEL）	¥28,000	¥20,500
◆CZ-213MSDMUSICPRO68K	¥18,800	¥13,200
◆CZ-214MSDSOUNDPRO68K	¥15,800	¥11,300
◆CZ-215MSDSamplingPRO68K	¥17,800	¥12,500
◆CZ-220BSDDATAPRO68K	¥58,000	¥40,000
◆CZ-225BSV Multiword Ver.2.0	¥32,000	¥23,000
◆CZ-243BSDCYBERNOTEPRO68K	¥19,800	¥15,000
◆CZ-247MSDMUSICPRO68K（MID）	¥28,800	¥20,500
◆CZ-249GSDCANVASPRO68K	¥29,800	¥22,000
◆CZ-251BSDHyperword	¥39,800	¥29,400
◆CZ-253BSDCARDPRO68KVer.2.0	¥29,800	¥22,700
◆CZ-257CSDCommunicationPRO68KVer.2.0	¥19,800	¥15,300
◆CZ-258BSDTeleportionPRO68K	¥22,800	¥16,900
◆CZ-261MSDMUSICstudioPRO68KVer.2.0	¥28,800	¥21,200
◆CZ-263GWDEasypaintSX-68K	¥12,800	¥9,800
◆CZ-264GWD Easydraw SX-68K	¥19,800	¥15,300
◆CZ-265HSDNewPrintShopVer.2.0	¥20,000	¥15,400
◆CZ-266BSDPressConductorPRO68K	¥28,800	¥22,000
◆CZ-267BSDCHARTPRO68K	¥38,000	¥29,800
◆CZ-272CWCCommunicationSX68K	¥19,800	¥14,500
◆CZ-275MWDSOUNDSX68K	¥15,800	¥11,500
◆CZ-284SSDOS-9/X68000Ver.2.4	¥35,800	¥25,600
◆CZ-286BSDBUSINESSPRO68KPopular	¥28,000	¥20,500
◆CZ-288LWD開発キット（workroom）	¥39,800	¥29,700
◆CZ-290TWD SX-WINDOW ディスクアクセサリー集	¥14,800	¥11,500
◆CZ-294SS(5")/SSC(3.5") SX-WINDOW Ver.3.0	¥19,800	¥15,200
◆CZ-295LSD C-Compiler PRO68K Ver.2.1 NEW KIT	¥44,800	¥32,500

☆ゲームソフト25%OFF OK!!（一部ソフト除く）

注目!! 冬のボーナス一括払い手数料（金利）無料（平成5年10月末／11月末／12月末のいずれかをご指定ください。）

The スーパーファミコン 編集部責任編集

スーパーガイド・シリーズ

好評発売中!

●全キャラクターの対戦攻略を完全マスターしよう!

ストリートファイターⅡターボ スーパーガイド

A5判・定価650円
大ヒット中の「ストリートファイターⅡターボ」の完全攻略本。
全12人の登場キャラクター別にそれぞれの対戦攻略を徹底ガイド。
また、新必殺技など基本技以外の決め技を大紹介。

●シリーズ全4本完全マップつき徹底攻略!

スーパーマリオコレクション スーパーガイド

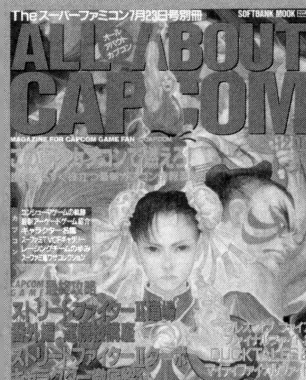
A5判・定価890円
人気ソフト「スーパーマリオコレクション」の完全攻略ガイドブック。
「スーパーマリオブラザーズ」・「スーパーマリオブラザーズ2」・「スーパーマリオブラザーズ3」・「スーパーマリオブラザーズUSA」という4本の大人気ソフトをマップ付きガイドで徹底攻略。

好評発売中!

●カプコン・ガイドブックの決定版!
Theスーパーファミコン7月23日号別冊

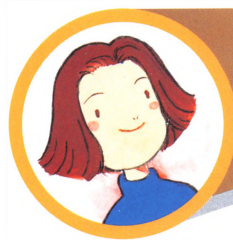
ALL ABOUT カプコン

カプコンのすべて

AB判・定価1200円
「ストリートファイターⅡターボ」ほか人気のカプコンSFCゲーム、
FCゲームの総ガイド、カプコンゲーム史、キャラクター図鑑、
裏技大全などカプコンの魅力100%

SOFT BANK ソフトバンク株式会社
出版事業部

■定価はすべて税込です。
■お求めはお近くの本屋さんで。

パソコン/ワープロ通信ネットワークサービス

J&P HOT LINE

今回は、パソコン通信ネットワークが産声をあげたころから通信をされているSHUさん。まさにパソコン通信のパイオニア的ユーザーであり、自らパソコンネットワークを運営されている本格派でもあります。HOTＬＩＮＥをはじめ、他ネットにも積極的にアクセスされているSHUさんに貴重なご意見をうかがいました。

＝基本データ＝

■使用機種名：本体：元祖X68000(CZ-600C)
■所有周辺機器：ハードディスク(CZ-620H)
プリンタ(CZ-8PC2)
カラーイメージユニット(CZ-6VT1)
モデム(MD96FB5V)
■使用開始時期：1987年頃

■X68000を選んだ理由は？

国産初のOSを搭載したパソコンであり、標準で68000CPUによるセグメントを意識せずにプログラミング出来る環境が整っていたからです。

■主にどんな用途に使われていますか？

パソコン通信……自らホストプログラムを作成し、時間限定ですが、BBS「SHU-NET」を運営しています。

プログラミング…もっぱら小さなユーティリティが主ですが、BASIC、アセンブラ、C言語を使い分けています。

■X68000を使っての満足感、楽しい部分は？

キーボードでほとんどの制御ができること。パソコンがテレビへ変身するし、フロッピィははきだすし、リセットはできるし、やりたいことが即行動に移せるマシンだと思います。

■X68000、こうだったらいいのになぁと思うことは？

X68000が登場したときのスペックは、他のパソコンと比べものにならないほど高機能だったが、今日では目新しくもなくなっている。CD-ROM搭載の機種があってもいいのではないでしょうか。

■その他、面白い使い方を教えて下さい。

無印X68000にしかないリモート端子（外部パワーオン）を利用して、雨を検知する検知器を自作・接続。雨が降り出すと自動起動して「雨が降ってきましたよ！」と知らせてくれます。

■HOTLINEに入会したのはいつごろですか？

今では考えられない300bpsで実験稼動の初期段階からの参加。

■よく利用しているコーナーは？

データベース／週刊クリッピングニュース。毎週金曜日の夜にのぞくコーナーで、１週間の出来事をおさらいします。

■HOTLINEを何に活用されていますか？

新製品、イベントなどの情報収集、オンラインショッピング。オンラインショッピングでは、インクリボンやフロッピィのまとめ買いをしています。

■X・MZユーザーに知らせたいHOTLINEのコーナーは？

ショッピング・プロムナード／今週の掘り出し市。あっという掘り出しものを安く手に入れることができるかも。

■あなたにとって、J&P HOTLINEとは？

幼なじみ。パソコン通信誕生から一緒に関っているから。

J&P HOT LINEへの
ご入会はスタータキットで。

お求めは、下記のお店へ。又は現金書留にて、¥3,000＋¥90(消費税3%)＝¥3,090を事務局までお送り下さい。
すぐにスタータキットをお送りします。

お問い合わせは
〒556 大阪市浪速区日本橋西1-6-5 上新電機株式会社
J&P HOTLINE事務局宛　TEL.(06)632-2521

スタータキットのお求めはJ&P各店でどうぞ。

店名	電話
渋谷店	☎(03)3496-4141
町田店	☎(0427)23-1313
八王子店	☎(0426)26-4141
立川店	☎(0425)36-4141
三鷹店	☎(0422)31-6251
横浜店	☎(045)313-6711
本厚木店	☎(0462)25-5151
津田沼店	☎(0474)72-5211
越谷店	☎(0489)66-1221
焼津インター店	☎(054)626-3311
にいがた1ばん館	☎(025)241-3711
富山店	☎(0764)22-5033
金沢店	☎(0762)91-1130
寺地店	☎(0762)47-2524
大須店	☎(052)262-1141
テクノランド	☎(06) 634-1211
メディアランド	☎(06) 634-1511
コスモランド	☎(06) 634-3111
U.S.LAND	☎(06) 634-1411
ビジネスランド	☎(06) 348-1881
高槻店	☎(0726)85-1212
くずは店	☎(0720)56-8181
千里中央店	☎(06) 834-4141
摂津富田店	☎(0726)93-7521
寝屋川店	☎(0720)34-1166
枚方バイパス店	☎(0720)48-1211
藤井寺店	☎(0729)38-2111
岸和田店	☎(0724)37-1021
さんのみや1ばん館	☎(078)231-2111
西宮店	☎(0798)71-1171
伊丹店	☎(0727)77-5101
姫路店	☎(0792)22-1221
京都寺町店	☎(075)341-4411
京都近鉄店	☎(075)341-5769
大久保バイパス店	☎(0774)44-1211
和歌山店	☎(0734)28-1441
和歌山南店	☎(0734)25-1414
学園前店	☎(0742)49-1411
奈良1ばん館	☎(0742)27-1111
新大宮店	☎(0742)35-2611
郡山インター店	☎(07435)9-2221
田原本店	☎(07443)3-4041
熊本店	☎(096)359-7800

Oh!X 10月号 1993年10月1日発行(毎月1回1日発行) 通巻138号 昭和58年11月2日第三種郵便物認可 ソフトバンク株式会社発行 Printed in Japan 特別定価800円(本体777円)

T1002179100801 雑誌 02179-10